BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF INFORMATION TECHNOLOGY
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF INTELLIGENT SYSTEMS
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

SECURITY AND USABILITY OF PASSWORD MANAGERS

BEZPEČNOST A POUŽITELNOST SPRÁVCŮ HESEL

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR **MARTIN HADERKA**
AUTOR PRÁCE

SUPERVISOR **Mgr. KAMIL MALINKA, Ph.D.**
VEDOUCÍ PRÁCE

BRNO 2022

Brno University of Technology
Faculty of Information Technology

Department of Intelligent Systems (DITS) Academic year 2021/2022

Bachelor's Thesis Specification

Student: **Haderka Martin**
Programme: Information Technology
Title: **Secure and Usable Password Manager**
Category: Security

Assignment:

1. Get familiar with existing password managers and coding best practices related to password storage.
2. Discuss security properties of existing password managers (focus also on the usable security).
3. Discuss the features of the ideal password manager.
4. Design and implement multiplatform password manager.
5. Conduct testing, focusing on security and performance parameters.

Recommended literature:

- ByPass: Reconsidering the Usability of Password Managers. Elizabeth Stobert, Tina Safaie, Heather Molyneaux, Mohammad Mannan, Amr Youssef. EAI International Conference on Security and Privacy in Communication Networks (SecureComm), Oct 21-23, 2020.
- Password Managers: Under the Hood of Secrets Management (available online: https://www.ise.io/casestudies/password-manager-hacking/)
- Sean Oesch and Scott Ruoti. 2020. That was then, this is now: a security evaluation of password generation, storage, and autofill in browser-based password managers. In Proceedings of the 30th USENIX Security Symposium. USENIX.

Detailed formal requirements can be found at https://www.fit.vut.cz/study/theses/

Supervisor: **Malinka Kamil, Mgr., Ph.D.**
Head of Department: Hanáček Petr, doc. Dr. Ing.
Beginning of work: November 1, 2021
Submission deadline: May 11, 2022
Approval date: November 3, 2021

Abstract

Login credentials are an indivisible part of internet users. Credentials are a mechanism that provides proof of an authorised person and, at the same time, prevents personal data abuse by unknown users. The number of passwords that a user has to remember has increased significantly in recent years. There are tools (password managers) available that simplify this problem and are easily accessible. However, the adoption of password managers is still weak.

The goal of this thesis is to identify problems that affect the usability of password managers and attempt to offer a solution to these problems. The analysis also includes an inspection of security and usability vulnerabilities that current password managers suffer from. In the last part, an example of password manager is designed and implemented, which provides important functions and should be accessible and easy to use by all Internet users.

Abstrakt

Přístupové údaje jsou nedílnou součástí uživatelů internetu. Jedná se o mechanizmus, který umožňuje prokázání se, že se jedná oprávněnou osobu a zároveň zabraňuje znežití osobních dat cizími uživateli. Počet hesel, který si dnešní uživatel musí zapamovat, se v posledních letech značně navýšil. Existují programy (správci hesel), které tento problém uživatelům ulehčují, a jsou zároveň snadno dostupné. Přesto je používá málo lidí.

Cílem této práce je zjištění problémů, které mají dopad na použitelnost správců hesel a zároveň se snaží nabídnout řešení těchto problémů. Součástí analýzy je i nalezení bezpečnostních a použitelnostních nedostatků, kterými aktuálně správci hesel trpí. V poslední části je navržena a implementována ukázka správce hesel, který nabízí důležité funkce, je dostupný a snadno použitelný všemi uživateli internetu.

Keywords

Authorization, password managers, usability, security

Klíčová slova

Autorizace, správci hesel, použitelnost, bezpečnost

Reference

HADERKA, Martin. *Security and Usability of Password Managers.* Brno, 2022. Bachelor's thesis. Brno University of Technology, Faculty of Information Technology. Supervisor Mgr. Kamil Malinka, Ph.D.

Rozšířený abstrakt

Přístupové údaje jsou nedílnou součástí uživatelů internetu. Jedná se o mechanizmus, který umožňuje prokázání se, že se jedná oprávněnou osobu a zároveň zabraňuje zneužití osobních dat cizími uživateli. Počet hesel, který si dnešní uživatel musí zapamatovat, se v posledních letech značně navýšil. Lidé si proto stále častěji volí stejná, nebo velmi podobná hesla pro více účtů. Tento zlozvyk je vystavuje riziku, kdy odcizení hesla z jednoho účtu otevře přístup k dalším účtům. Uživatelé si většinou toto riziko uvědomují, ale neřídí se pravidly volby bezpečného hesla, které toto riziko eliminuje, jelikož je velmi obtížené si zapamatovat větší množství hesel, hlavně v případě starších uživatelů.

Řešením tohoto problému je použití aplikací zvaných správci hesel, které tento problém uživatelům ulehčují, a jsou zároveň snadno dostupné. Správci hesel nabízí mnohem více než jen snadné uložení hesel, většina nabízí i bezpečné uložení jiných věcí, jako například čísla kreditních a identifikačích karet. Mnoho správců hesel i usnadňuje práci s prohlížečem a to použitím rozšíření do prohlížeče. Při použití správce hesel si uživatel musí pouze pamatovat jedno hlavní heslo, pomocí kterého odemkne databázi všech uložených hesel. Uživateli v ten moment stačí otevřít správce hesel a ten za ně vyplní hesla na webové stránky, které aktuálně prochází. Správce hesel nabízí i další bezpečnostní výhody, například kontrolu URL adresy, mazání hesla ze schránky, atd. Samotné použití správce hesel však nezajišťuje, že si uživatel zvolí unikátní hesla na různé služby. Správci hesel proto nabízí možnost generování unikátního hesla, které poté uživatel použije při registraci nového účtu. Uživatel, ale musí o této možnosti vědět a měl by ji použít. Generování hesel má však své nedostatky, kdy různé webové služby vyžadují jinou kombinaci malých a velkých písmen, čísel a speciálních znaků. Během posledních let došlo k zvýšení uživatelů správcvů hesel, avšak je v celku používá málo lidí. Důvodů, proč se jim uživatelé vyhýbají je mnoho.

Cílem této práce je zjištění problémů, které mají dopad na použitelnost správců hesel a zároveň se snaží nabídnout řešení těchto problémů. Součástí analýzy je i nalezení bezpečnostních a UX nedostatků, kterými aktuálně správci hesel trpí. V poslední části je navržena a implementována ukázka správce hesel, nabízí důležité funkce, je dostupný a snadno použitelný všemi uživateli internetu.

Security and Usability of Password Managers

Declaration

I hereby declare that this Bachelor's thesis was prepared as an original work by the author under the supervision of Mgr. Kamil Malinka, Ph.D. I have listed all the literary sources, publications, and other sources that were used during the preparation of this thesis.

.....
Martin Haderka
May 5, 2022

Acknowledgements

I wish to acknowledge my supervisor for his guidance throughout the assignment, my family for their support, and everyone who participated during the testing of the custom implementation of password manager; without those, this work would not have been possible.

Contents

Chapter 1

Introduction

In the early days of computers, passwords and login prompts were generally not necessary to use a computer. Computers used to be large room-size machines used mainly for calculation purposes. Passwords and user accounts were introduced at MIT in the CTSS project[14]. A big jump in password usage (in the form of an identification technique) occurred with the rise of the Internet. Login items became an everyday part of Internet user's life. A 2014 study has found an average Internet user types an average of 23 passwords per day[21]. The overall average account count, according to a post from 2017 in LastPass' blog, is 191.[1] This number continues to increase as more services become available.

It is important to keep in mind that using a password does not provide outright protection, the password needs to fulfil several characteristics to be considered "secure". There are several standards that a user can follow, published by organisations like OWASP[2] and NIST[3], but it would be excessive expecting every user to check their passwords against current security standards. Many sites may enforce different password policies, therefore, an easier way to pressure a user to create and use a secure password is needed. To further secure login, multifactor authentication (MFA) techniques, such as Time-based One-Time Passwords (TOTP) or a hardware authentication devices, may be required as additional protection to the sign-in process.

Generally, security enhancements often come with usability downgrades. Finding a balance between these two factors may be an impossible task. This also applies to the usage of passwords today[16]. Using a complex, unique, and frequently changed password is secure, but is not ideal from a usability point of view. Users have trouble remembering complex passwords and drift to less secure behaviour, either in the form of password reuse or in less complex password usage. This insecure habit increases the possibility that passwords are stolen and misused if the attacker can reuse one password on all services.

To ease the user burden, there is a group of computer programs called password managers. Their main function is to store all user passwords in one place under one complex password. Using a password manager has further advantages, which will be discussed in the following sections. Password managers are freely available and there has been an uptrend in pass-

[1] Available at: https://blog.lastpass.com/2017/11/lastpass-reveals-8-truths-about-passwords-in-the-new-password-expose/ Accessed on 2022-04-30
[2] More about OWASP at https://owasp.org
[3] More about NIST at https://www.nist.gov

word managers' adoption in the past years, but the overall adoption has been quite low. One study, from 2009, shows that only 1% of the respondents used a password manager[9]. Other survey, from 2017, shows that 55% of the respondents used a password manager[5]. The results mainly depends on demographics, technical skills and age. We can see that the users count is rising, but still not everybody uses a password manager. Therefore, I find it important to discuss challenges that may prevent users from using a password manager.

Following the topic of cybersecurity, massive data breaches have become a regular occurrence in recent years[4]. Therefore, the use of a strong password resistant to brute-force and dictionary attacks is a requirement. Password managers can help us with this effort with their password generator feature.

The main goal of this thesis is to analyse and categorise existing and popular password managers and create an overview of security implementations and standards of authentication and secure data storage. Another goal is to identify the key features of a password manager, describe an ideal password manager, and implement it.

Chapter 2

Password, Password Manager and Security

In this chapter, a deeper look at passwords, password managers, and related security is concluded. Passwords, which are used every day, can be divided into certain types, depending on the used characters. To ensure the user's secure behaviour, a way to create a new secure password and an overview of possible attacks and password thefts are provided. Password managers are divided into categories according to the way they are used, with a description of basic functionality in terms of usability and security and issues that a user may encounter when using a password manager. On a general security topic, a definition of individual measures can be found, that not only password managers, but all Internet services, in general, should comply with.

2.1 Password

A password login is a method of authentication, generally in the form of several characters, including letters, numbers, and special symbols. It is used to confirm the user's identity on websites, usually in combination with other identifiers such as email or username. Passwords may also come in several other forms. The system is based on the belief that only the authenticated user knows the password corresponding to the email or username. The security of a password depends on the length, sequence, type, and occurrences of used characters, and on the current computational power which may be used to guess the secret password. The system that authenticates the user must also follow several security measures to ensure that the authentication process is valid.

2.1.1 Password Forms

2.1.1.1 PIN

Personal identification numbers (PINs). PINs are used to secure access to applications, either in combination with a token for two-step identification (for example, for cash dispensers), or as an onestep authentication (for example, mobile phones, home burglar alarms).[22] A password consisted only of numbers. It is commonly used for credit card authentication. The length of the PIN or passcode is short, usually between 4 or 12 characters, so it can be easily remembered.

2.1.1.2 General password

Passwords consist of strings of alphanumeric characters. To prevent cracking attacks, security experts advise that users must have strong passwords (a nonmeaningful string of characters drawn from a large character set, mixing letters, numbers and symbols, and upper and lower case).[22] Password (general password) is the most used type. It also usually contains special characters. The requirement of numbers and special characters causes that the password is hard to remember, but makes it resistant to dictionary attacks if it is chosen correctly. The suggested length of the password is rising as the computational power grows, therefore the password could not be easily cracked by brute-force attacks.

2.1.1.3 Passphrase

A passphrase is a sequence of several easily remembered words that can be used for authentication. The passphrase is usually longer (consisted of more words) than the general password, to enhance its durability against brute-force attacks. It is commonly used to derive an encryption key, for example, in cryptocurrency wallets.

2.1.2 Password Recommendations

Generally, it is required that the password is unique, complex, and secure. Uniqueness can be achieved by not reusing one password on multiple accounts and not using commonly known passwords, such as "1234", "password", "qwerty", etc. Complexity may be achieved by using all types of characters, i.e., numbers, special characters, lowercase and uppercase letters, all with appropriate length. Users should not write passwords on a piece of paper or in unencrypted text files on a computer. Users tend to choose a password that is easily remembered and personal, such as names of relatives or pets, dates of birth, names of favourite places, etc. The combination may seem complex and unique, but the password may be vulnerable to social engineering attacks. To avoid these attacks, users should rather use a password manager, which can generate a random password with the desired length and complexity, and store the password in encrypted storage.

2.1.3 Password Attacks

2.1.3.1 Password Capturing

Not only does the user need to choose a secure password, but the service must transmit the password and authenticate correctly and securely. Websites with login forms should not use a simple HTTP protocol that does not encrypt transmitted data between the client and the authenticating server, the password is then sent in plain text and may be captured on the way to the server. It is recommended to use Transport Layer Security (TLS) with HTTP, which provides bidirectional encryption between the client and the server, thus avoiding the threat of capture.

2.1.3.2 Database Leak

An authenticating server should not store passwords in plain text in case of a database leak. The authentication process should only compare the hash of the password, therefore even the server owner should not be able to see the user's password. The hashes do not provide bulletproof security, since they may be sensitive to rainbow attacks, a dictionary

of known hashes. A salt (a random string of characters) could be added to the password before hashing for further security. User data stored in the database should be encrypted with an encryption key which is derived from the user's master password.

2.1.3.3 Dictionary Attack

Unlike checking all possibilities using brute force attack, the dictionary attack tries to match the password with most occurring words or words of daily life usage.[19] A list of the most used passwords, which may be also used in an attack where the user's password is being guessed. Users should avoid common passwords for better security.

2.1.3.4 Phishing Attack

A type of attack, in which a person may be fooled into opening a fake website and be prompted to log in there. The attacker redirects the user to another website e.g. "www.yah0o.com" whose interface is similar to that of the original website to disguise the user.[19] The password manager may help with this, since it would not fill in or prompt to fill in when the website's URL does not exactly match, and therefore the users may realise that they are on a fake website.

2.1.4 Conclusion

We looked into password types and the places they are used. In terms of password recommendation, user should always generate a unique, long, and complex password for every account, password managers can help with this requirement. The user should be aware of possible password theft and should notice some warning signals to prevent a potential attack.

2.2 Password Managers

Simplified definitions introduce "password managers" as computer programs that are designed to create, store, and use passwords on websites or in other software. Whether a password manager can offer additional functionality depends on where and how it is used. To expand the simplified definition, most password managers allow the user to store other items such as credit card information, ID card numbers, notes, and files in encrypted storage. Another feature is the possibility to generate a new password with the desired complexity. Complexity can be defined and configured by setting a minimal or maximal length and other parameters, such as the minimal count of numbers, letters, and special characters, to strengthen the generated password and to fulfil a website's password policy. There are a few types of password managers. An average user will mainly use a browser or a cloud-based password manager. Many of those are available for free, but some password manager creators also offer paid versions with additional functionality.

Password managers can be classified into the following categories: browser, cloud, local, and hardware password managers. Each category is specified in the following sections. As stated above, users have many options to choose from. In most cases, they do not need to download any additional software, because many web browsers come with a built-in password manager. The popularity of cloud-based password managers is increasing, for

example, LastPass's user count was 16.8 Million in 2018[1] and 25 Million in 2020[2].

This raises a security question of password managers. A survey has shown that 65% of Americans do not trust password managers[3]. Password managers store a user's private information in a database on a remote server or locally, depending on the type of password manager. It makes password managers a high reward target for a potential attacker in the case of a successful security breach and allows them to steal personal credentials for multiple platforms in one sweep. Therefore, the security of password managers is one of the key topics in general and this thesis.

Since the variety of password managers is high, instead of focussing on concrete implementations, I will focus on each type to determine what are the advantages and disadvantages of each and only within each category, I will examine a concrete implementation as a representative of the type. Also, what limitations and features each type comes with. Many of the password managers can be assigned into more categories, depending on the features their users will activate upon subscription.

2.2.1 Types of Password Managers

2.2.1.1 Browser-based

A browser-based password manager is a password manager that comes built into a website browser. The user does not need to download any additional extension or application. If this feature is activated, which is in the most cases by default, the browser will capture and save each credential that the user types into a website's registration or login form. Next time, when the corresponding site is visited, the browser will automatically fill the user's credentials, simplifying and speeding up the login process. This type of password manager is usually the first one that the user comes in touch with and is convenient to use. Furthermore, this type of manager typically avoids users questioning their security, as the password database itself is stored locally if multiple device synchronisation is not turned on. In most cases, there is no need for additional actions from the user, e.g., setting up a master password, which is needed in other types of password managers. However, it comes with many limitations. This type of password manager does not provide multiple device synchronisation by default, with exceptions, e.g., Google Chrome provides a possibility to synchronise stored passwords through a user's Google account.

2.2.1.2 Cloud-based

The key property of cloud password managers is that they store passwords in cloud storage, i.e., a remote server. Meaning that the passwords are not stored locally on a computer. This allows for easy synchronisation across multiple devices, but each device needs a corresponding application and a browser extension to connect to the remote password database. One of the disadvantages is the need to create an account under which the remote database is stored. The account's password, in most cases referred to as the master password, is

[1] Celebrating 10 Years of LastPass. Available at: https://blog.lastpass.com/2018/07/celebrating-10-years-lastpass/. Accessed on 2021-12-13.

[2] LastPass Celebrates 25 Million Users. Available at: https://blog.lastpass.com/2020/09/lastpass-celebrates-25-million-users/. Accessed on 2021-12-13.

[3] 65% of people don't trust password managers despite 60% experiencing a data breach. Available at: https://www.passwordmanager.com/password-manager-trust-survey/. Accessed on 2021-12-13.

used to encrypt and decrypt the items stored in the database. They usually provide other features, such as notification of a web service breach, for example, Yahoo security breach in 2013, when 3 billion user accounts were affected.[4]. And a warning of password reuse, easily guessable or common password usage. They also come with advanced password generator features. The main disadvantage is that personal data are stored in a remote location, only protected by the master password. If the master password is easily guessed, stolen, or otherwise disclosed, all passwords may be compromised. In such a case, the attacker's knowledge of this password creates a doorway to all stored passwords, as well as knowing the services in which the said passwords were used. Therefore, using a secure, not easily guessable master password is a must, and most cloud password managers require a secure master password by placing restrictions on the minimal length and complexity of the master password when the account is created. There have also been some breaches of remote databases from cloud password managers, for example, LastPass breach in 2015 [5]. In those breaches, an attacker gained access to the corresponding servers. Even if the database, sometimes called a vault, is stolen by the attacker, the data in it are not readable unless the attacker gets the user's master password and used cryptographic security in hand as well. Data decryption without having the master password requires a high amount of time and computation resources, hence giving users the time necessary to change all passwords before they can be abused by the attacker, assuming the breach is detected and disclosed early enough. The companies behind cloud password managers mostly provide a subscription-based model (monthly or anually paid) to unlock premium features (e.g., the password health check, which is checking the strength of the password). For enterprise users, their paid model provides another services, such as help desk support.

2.2.1.3 Local-based

The main advantage of local password managers is that the password database is stored locally. Therefore, stealing passwords requires physical access to the device disk. Many password managers allow storing user's items locally while providing the same functionality as if they were, e.g., cloud-based, except for easy multiple device synchronisation.

2.2.1.4 Hardware-based

HW password managers are a type of password manager stored on an external physical device (mainly USB flash drives). Into this category, we can include password managers that have Multi-factor authentication (MFA) with credentials stored on a USB drive. The advantage of such password managers is that the database is easily transferable and stored only in one location (this category is a special type of local password manager). They mostly lack other features than storing and providing passwords, but it highly depends on the application.

Cryptocurrency hardware vaults can be included into this category, which stores keys for a cryptocurrency wallet, e.g., hardware wallets from companies like TREZOR, Ledger and CoolWallet.

[4]BBC News: Yahoo 2013 data breach. Available at: https://www.bbc.com/news/business-41493494. Accessed on 2021-12-13.

[5]LastPass Hacked – Identified Early & Resolved. Available at: https://blog.lastpass.com/2015/06/lastpass-security-notice/. Accessed on 2021-12-13.

2.2.1.5 System-wide

This category describes the password managers that are built into the operating system. They provide local password storage of applications and websites that were visited by the signed-in user. One of the advantages is that users do not need to download and install third-party applications to maintain their passwords. They cannot usually synchronise or export stored passwords to another operating system, or are limited only to OS-based applications. Examples of such OS-based password managers are Apple Keychain and Windows Vault.

2.2.2 General Features

2.2.2.1 General Functionality

A password manager manages (i.e., create, read, update and delete) user's passwords and other personal information in a secure location. Management is possible with a client application and browser's extension. Secure storage is achievable by encrypting the user database with a strong encryption function, such as the Advanced Encryption Standard, and with an appropriate setting. Cloud-based password managers operate on a server, which must ensure secure user authentication, therefore correct usage of hashing functions, key derivation functions, secure connection, and database distribution is required.

2.2.3 Additional Features

2.2.3.1 Password Capturing

Password managers with extensions should be able to monitor and capture user login details in the browser and offer to store new login credentials or to update an existing one. This feature simplifies password management, so that the user does not need to manually add or edit passwords.

2.2.3.2 Password Generator

Making up a unique and random password can be difficult for most users. Most users choose either a single password that they use on multiple sites or a password based on a specific format. Password managers, which offer the ability to generate a random password, greatly simplify this step by generating a random and unique password for the user in a second. An issue may arise if the pages require a certain number of lowercase and uppercase letters, numbers, and special characters. Therefore, the password manager should offer the option of specifying these requirements before generating a new password. This ensures that the password is valid, unique, and random.

2.2.3.3 Password Filling

If the user visits a page for which he already has a saved password. The password manager should offer to fill in the login details, either automatically or, if necessary, after using a keyboard shortcut. This also offers a certain level of security, where if a user appears on a fake page, for example, due to a phishing attack, the password manager will refuse to fill in the password automatically if the URL address does not match the address stored with the given login details. The password manager cannot fully secure data from being entered on a fake page if the user copies the password from the application, but forces the user to

raise attention when the password manager does not fill the credentials automatically or on request.

2.2.3.4 Password Strength Check

Passwords selected by the user or not generated by the password manager may not have sufficient length or combination of characters to be considered secure. The password manager was supposed to show the user the strength of the security of such a password and alert the user to change the password in case of a wrong choice.

2.2.3.5 Dictionary Check

This feature is useful in situations where the user chooses the password by himself. Resulting in the risk of being among the commonly used passwords. Statistics of the most used passwords are published every year. For example, according to statistics from Nordpass[6], in 2020 the most used password was "123456". These lists of known and frequently used passwords can be misused for an attack. In any case, the user should avoid such passwords or its variants, and the password manager should detect such passwords and prompt the user to change them.

2.2.4 Problems of Password Managers

2.2.4.1 Database Leaks

The most commonly used category of password managers, cloud-based password managers, stores user data in remote storage. This option offers the advantage of easy password synchronisation among multiple devices. The user does not own or have physical access to the server on which his data are stored. The server owner takes over the data protection and, in case of data leakage from the database, the owner must inform all users of the given password manager. However, a leak does not in itself mean that user passwords are stolen in an unsecured form, if stored correctly, only the password hashes used for authentication and encrypted data will leak. With the correct use of strong hashing and encryption functions, an attacker will not be able to obtain data in a readable form without knowledge of used security and without using a lot of computational complexity.

2.2.4.2 Memory Leaks

A risk exists also on the client's side if the application or browser extension mishandles the memory and leaves passwords stored in a readable and insecure form. To manipulate the password, for example, copy and display the password, it is necessary to display the password in an insecure form. The application should then delete the password from the memory after a certain time, or after locking or logging off, to prevent possible theft. The theft can occur if the user has the wrong software installed on their computer or phone that can steal data from memory. Such attacks are usually targeted. To avoid this attack, the user should not install applications from unverified sources and could use antivirus software that may detect the application with bad intentions.

[6]Top 200 most common password list 2021, available at: https://nordpass.com/most-common-passwords-list/. Accessed on 2022-02-23.

2.2.4.3 Clipboard Leaks

If the user copies the password from the password manager to the clipboard, the application should delete the copied password from the clipboard after a certain time or after locking or logging off. Again, to prevent the possibility of an attack by a malicious application that can scan the data stored in the clipboard. Some users use an advanced clipboard manager that can view the history of copied data. Such clipboard managers also allow for the option of ignoring copied data from selected applications. The password manager should mark the data it inserts into the clipboard as confidential if the used system APIs allow it (such as the NSPasteboard API used on the MacOS operating system) so that these applications will not remember the copied data from the password manager.

2.2.4.4 Usability Difficulties

The password manager should, by default, set measures to protect user data as much as possible. A theft can happen when a user leaves an unlocked computer in public place (e.g., in library) therefore, the application should automatically lock after certain time to prevent this. The main elements of safety in terms of usability are the following:

- Automatic locking: the application should not hold any sensitive data in its memory when locked. Unlocking is enabled, for example, by a PIN code or biometric login.
- Automatic logout: the application does not keep any data in memory, and at the same time a master password and server-side authentication are required to unlock it.
- Automatic clipboard deletion: after a certain time, the password manager should delete the copied password from the clipboard.

All options should be configurable for each user, but should be alerted of possible danger when changing the default settings.

2.2.5 Conclusion

Password managers differ in some parts, mainly in the environment in which they are used. On the other hand, their basic functionality is similar across all of them. Advanced features, such as password generators or a security indicator for already existing passwords, can improve a user's online behaviour. It is important that users are not exposed to additional requirements that may prevent users from using the password manager.

2.3 Security

To provide secure data storage, password managers must comply with the latest security recommendations. The field which studies information security and secure communication is called cryptography, which applies mathematical techniques and implements them in computer science to ensure better security or analyse potential threats, e.g., time to crack encrypted data. While password managers store confidential data, they must use techniques such as hashing and encrypting, to ensure users' data security and validity. The usage of these techniques does not provide protection right away, other measures must be implemented, such as memory scrubbing, clipboard deletion, and automatic logout. In addition,

an user error must be eliminated by explaining potential threats and the need for secure password usage and general secure behaviour on the Internet.

2.3.0.1 Database Security

Password managers should be designed so that the database owner will not be able to view user data in an open format. This can be achieved in such a way that the key that encrypts and decrypts user data is generated by the user's client application and the data are encrypted before they are sent to the server's database.

2.3.0.2 Hash Functions

Hashing is the transformation of a string of characters into a fixed length value or key that represents the original string.[23] The hash function must follow two properties, it must be one-way, so it is impossible to retrieve the original value when given the hash, and it is collision resistant, meaning that two different inputs cannot generate the same value.[18]

Hash functions, especially collision-resistant hash functions, are used for storing login passwords. When the password is hashed, the server owner or a potential hacker will not be able to see the original password, and only hashes are compared when the user tries to authenticate.

However, there is a way to retrieve the original value. Hackers may use a table of precomputed hashes, i.e., a rainbow table. A way to prevent a rainbow table attack is to modify, usually append, the original value with random characters, usually called salt, which will prevent the use of rainbow tables.

Recommendation of hash functions depends on the final usage and on the computer's computational power increase, e.g., MD-5, a widely used hash function, is not recommended anymore. Several organisations publish a recommendation for the use of hash functions, e.g., NIST recommends SHA-224 and higher for use within the federal government[3]. This could also be applied to password managers, as "military grade" security is expected.

2.3.0.3 Symmetric Encryption

It is a way to encrypt the original message with a key and decrypt the original value with the same key. The password in symmetric encryption is the same for encryption and decryption, where both the sender and the receiver need to share the password. Encryption is used everywhere to support user security.

In password managers, the data stored in the database must be encrypted, as well as the data transmission between the server and the client application.

As with hash functions, the recommendations for symmetric encryption changes if a potential threat is found or if the computational power increases, for example, a Data Encryption Standard (DES), which was often used, is not recommended anymore, as it does not provide sufficient protection. In NIST guidelines[3] Advanced Encryption Standard (AES)[24] replaced DES as a suit-full encryption standard.

The widely used encryption is the Advanced Encryption Standard (AES)[24]. AES uses 128, 192, or 256-bit key size, the key length may be a security factor for several applications.

2.3.0.4 Asymmetric Encryption

Asymmetric key encryption helps solve the problem of symmetric key exchange. Instead of one key (as a password) as in the symmetric encryption, two keys are used: private and public. It is important to generate both keys, as they are in fact two passwords, each working only in a single direction. The public key is then used to encrypt any data, and the private key is used to decrypt the data or vice versa. Asymmetric-key algorithms are used to provide identity, integrity, and source authentication for digital signature and to establish cryptographic keying material using key agreement and key transport algorithms. These algorithms tend to be much slower than symmetric key algorithms[3] Therefore, they are not used for large data encryption, but can still play a role in the transport of the symmetric key, for example, in the TSL/SSL protocol.[20]

2.3.0.5 Digital Signature

2.3.0.5.1 Message Authentication Code algorithm (MAC) A MAC algorithm and a cryptographic key are used to generate a MAC that can be used to ensure data integrity and source authentication. A MAC is a cryptographic checksum of the data that can ensure that the data have not changed since it was either saved or transmitted.[3]

2.3.0.5.2 Hash-based Message Authentication Code (HMAC) A HMAC is a standard MAC algorithm built with cryptographic hash functions in combination with a secret key.[3]

2.3.0.6 Key Derivation Function (KDF)

A KDF is a cryptographic algorithm that derives one or more secret keys from an input (e.g., a user password). User's passwords should not be directly used as cryptographic keys, therefore key derivation functions are used in password managers to get keys for vault encryption and decryption.

2.3.0.6.1 Password-Based Key Derivation Function (PBKDF) PBKDF are KDFs with iterative HMAC derivation to increase resistance to brute-force attacks, by increasing the computational time requirement of password derivation. The PBKDF output string is called the master password, which is used to encrypt and decrypt a user's vault with stored passwords.

$$mk = PBKDF_{(PRF,C)}(P, S, kLen),$$

where PRF is Pseudo-random Function, C is fixed iteration count, P is password, S is salt, $kLen$ is length of the master key.[15]

2.3.0.6.2 HMAC-based Key derivation Function (HKDF) HKDF follows the "extract-then-expand" paradigm, where the KDF logically consists of two modules. The first stage takes the input keying material and "extracts" from it a fixed length pseudo-random key K. The second stage "expands" the key K in several additional pseudorandom

keys (the output of the KDF).[12]

$$okm = HKFD(ALG, KEY, L, info, S),$$

where ALG is used hashed function, KEY is input key, L is length of output, $info$ is optional context and application-specific information, S is salt, okm output keying material (of L octets). Several vendors use HKDF to increase the entropy of an inserted key.

2.3.0.7 Random Bit Generation

Cryptography and security applications make extensive use of random numbers and random bits. In cryptography, random values are needed to generate cryptographic keys.[3] It is generally impossible to generate truly random numbers, Pseudo-Random Number Generators are used for this purpose, for even higher randomness with larger entropy Cryptographically secure pseudo-random number generators are designed.

2.3.0.8 Conclusion

The security level determines the sensitivity of the data stored on the server. Password managers should be as secure as possible, as the user stores very sensitive data on them. It is necessary to ensure that all parts of the system are properly designed and that they use only authenticated mechanisms for user authentication and data encryption. Hash functions provide us with secure authentication of the logged-in user. Encryption converts the data into unreadable form, and their reconfiguration is only possible on the basis of the decryption key. Each mechanism has its weaknesses, and, for example, with the help of a brute-force attack, the data can be read. Therefore, there are methods that can significantly increase the computational complexity of such an attack. If these mechanisms are used correctly, we can conclude that the system is safe.

2.3.1 Security and Usability

2.3.1.1 Security

Password managers, as any application that stores confidential data, should comply with the latest security recommendations. Stored data should be encrypted with a proper cryptographic function and except for local password managers, there is a need for a secure authentication process, especially in cloud-based password managers.

There may also be a dispute over whether open-source password managers are more or less secure than closed-source. Many may argue that a closed-source is better because a potential attacker has a hard time analysing the inner functionality of the targeted password manager, but “security in obscurity” is not ideal and does not enhance overall security. open-source password managers might be vulnerable to attacks, when a security issue is noticed in the publicly available source code, on the other hand, independent researchers and end users with appropriate knowledge can review any part of the code, making the password manager more secure.

2.3.1.2 Usability

Every system is only as secure as its weakest point. This also applies to the user who uses a password manager. Password managers may comply with the latest security recommen-

dations, but the user has to use the password manager properly to fully ensure the entire security.

The user interface must be attractive and convenient to the user, so there is no additional burden to the user when using a password manager. Users may give up using a password manager when the user interface is unappealing or complicated. At best, the password manager should actively help the user, for example, with an easy password generator, provide UI pop-ups or notifications when filling out the registration form, and easy password management, for example, when the user changes a password on a certain site, a box asking for password change approval in the database is appreciated.

Another way to enhance user security is offering a check of already stored passwords and their security with the recommendation of password change when any password is considered as weak or frequently used, i.e., is presented in common password lists.

2.3.2 Security Organizations

Several organisations act on cryptography and general security on the Internet. Organisations can be state-sponsored or private. The following 3 known organisations can be represented in terms of password security, and cryptography.

- The Open Web Application Security Project (OWASP) is a non-profit foundation that works to improve software security.[7]
- The National Institute of Standards and Technology (NIST) is a physical science laboratory and agency of the United States Department of Commerce.[8]
- The International Organisation for Standardisation (ISO) is an international standard-setting body composed of representatives from various national standards organizations.[9]

These organisations make various recommendations in the papers they publish. Some important papers will be described below.

NIST in Special Publication 800-63B: Digital Identity Guidelines[7] adequately describe how the verifying server should handle secrets and authentication and other security and usability aspects of the process. Regarding the thesis topic, several recommendations can be used in password managers' security, e.g., the recommendation of Password-based Key Derivation Function 2 (PBKDF2) and Balloon as suitable key derivation functions, since key derivation functions (KDF) take a password, salt, and cost factor as inputs and generate a password hash and increase the cost of an attack. The recommendation of the password format also appears, according to the paper, passwords shall be at least 8 characters long, and the verifier should be allowed to choose maximum of at least 64 characters long password. It is also recommended to avoid passwords obtained from previous breaches, dictionary passwords, repetitive or sequential characters, context-specific words, such as the name of the service, username, and similar. The paper also recommends normalisation of

[7]More about OWASP at: https://owasp.org
[8]More about NIST at https://www.nist.gov
[9]More about ISO at https://www.iso.org/

Unicode characters before hashing if they are allowed to be used in passwords. An interesting suggestion is the provision of guidance for the user, such as a strength meter for the password's security.

OWASP in Password Storage Cheat Sheet[10] recommends Argon2id, scrypt, bcrypt with a work factor of 10 and PBKDF2, the same as NIST recommends, with a high iteration count depending on the internal hashing algorithm.

Protection of sensitive information is described by OWASP in the OWASP Developer Guide[11] and Cryptographic Storage Cheat Sheet[12], and NIST in Special Publication 800-175B Revision 1: Guideline for Using Cryptographic Standards in the Federal Government[3]. It is recommended to use block-cipher algorithms, specifically AES[24] with a key of at least 128 bits, as a sustainable encryption algorithm.

Few guidelines are not specific, e.g., in an ISO/IEC 27001[11] specification follows "Password management systems shall be interactive and shall ensure quality passwords.". The concrete implementation is up to the designer and the security level of the final system.

2.3.2.1 Conclusion

With increasing digitalisation, it is necessary to ensure system security. Security can be viewed not only from the point of view of the security of stored and transmitted data, but also from the point of view of usability. Even in the case of the safest system, human error can occur, either due to ignorance of possible risks, but also an oversight can cause enough damage. To make it easier to implement security, there are organisations that actively monitor current threats, explore new security options, and issue security guidelines. Not all recommendations apply to every system, but when working with sensitive data, the system needs to be secured, those systems should meet the requirements of the organisations so that they can reduce the impact in the event of data theft.

[10] Available at: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html. Accessed on 2022-03-14.
[11] Available at https://andrewwhite.gitbooks.io/owasp/content/03-Build/0x11-Cryptography.html. Accessed on 2022-03-14.
[12] Available at: https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html. Accessed on 2022-03-14.

Chapter 3

Related work

Due to the current increase in password managers usage, several studies and articles have already been published on this topic. They vary mainly in depth and view of the analysis, but all share a similar main interest in usability and security issues. There is also a third group of studies that tries to present other authentication mechanisms that could substitute current password managers and provide solutions to problems which the current architecture of password managers is not able to solve, e.g., automatically fulfilling the website's password policy, when a new password is generated.

3.1 Security

Oesch and Ruoti[17] have studied the security of 13 browser-based password managers. The study shows a comparison of used encryption methods and resistance to known issues. They declared that the password managers' security had improved over the last 5 years, but they expressed a few vulnerabilities that need to be handled. They also inspected the password generator options, analysed millions of passwords generated by password managers, and reported issues with their results. In the future work section, they expressed a need for better browser support for password manager extensions and an issue with a different character set options in password generators, which could be standardized.

ISE researchers[10] studied 5 password managers that work on Windows 10. Their goal was to analyse the security practises of inspected password managers. They studied password managers in 3 states – not running (installed, but not opened after the last restart or shutdown), running & unlocked and running & locked. Their goal was to find out if password managers devalue memory and if passwords are visible after being typed or copied. They also looked at keylogger and clipboard sniffing defences, which all of the tested password managers succeeded in. They have found issues with memory scrubbing where passwords were left in visible form in cache files. They also provided suggestions for several guarantees which password managers should apply.

Luevanos et al.[13] have visited 3 open-source password managers - Passbolt, Padlock, and Encryptr. They discuss the difference between open-source and closed-source password managers, e.g., open-source password managers have better testing possibilities and modification, faster vulnerability fixes, but may lack security audits, when password managers are backed by small teams or companies. In their work, they analyse security flaws

and provide potential solutions. They also present their vision of an ideal password manager. In this study from 2017, the authors exposed the vulnerabilities of clipboard sniffing and keylogger attack in the Encryptr and Padlock password managers. Authors believe that security should be prioritised over usability, especially that password managers should demand strong master passwords to reduce their vulnerability to a brute-force attack. Furthermore, they should better prevent keylogger and clipboard sniffing attacks.

Gray et al.[8] have studied the security of several password managers. They have found that KeePass and RoboForm store the master password in readable form in system page files when a low amount of RAM is provided. KeePass also does not require master password input before database export, which can lead to the possible misuse of an attacker if control of the device is acquired.

3.2 Usability and Alternatives

Stobert et al.[25] had introduced a new password manager in their study, which would offer better usability than existing solutions. By creating an API that would stand between the user and the website, rather than extend user options. Their proposal had the same security benefits as traditional password managers by using strong encryption algorithms AES-256 and the PBKDF2 key derivation function and, at the same time, providing better usability. Their prototype in the form of the Google Chrome extension has shown positive user acceptance, but the authors faced the challenge of convincing existing websites to implement the provided API, so users can embrace every feature that the proposed password manager provides.

Fagan et al.[5] concluded a survey to investigate questions about the adoption and usability of password managers and the emotional response to the password manager of each participant. The responses to their survey were divided accordingly by type of respondents to password manager users and non-users and performed comparative analysis. Their findings concluded the following:

1. Password manager users have higher computer knowledge and computer security ability,
2. Password manager users find password managers more useful, convenient, and secure,
3. Non-password manager users mention possible security and usability issues of password managers as the main reason for not using a password manager,
4. Password manager users feel more secure when using a password manager to log in to a website

They concluded that there may be a lack of knowledge of password managers' theory. The explanation of the inner workings of the password manager may help improve adoption.

3.3 Conclusion

There is a lot of research on password managers, it contributes to the security of password managers. The researchers looked at password managers from different perspectives, from the used security, where they pointed out possible or existing security issues, but also from the point of view of usability, where they tried to find out what discourages users from using

password managers. Authentication with passwords is not ideal, especially when other options, such as single sign-on or magic link, are available. But everything has downsides and upsides, and passwords will still be around. The following chapters will look at the most used password managers, the goal is to find out which features may be considered as important and which should be in the ideal password manager.

Chapter 4

Comparison of Popular Password Managers

In previous chapters, we described general password managers' features and additional functions. General security techniques were also introduced. This chapter inspects currently popular password managers on the market. The following password managers: LastPass, Dashlane, 1Password, RoboForm 8 Everywhere, Sticky Password and Bitwarden as candidates for cloud-based password managers and Google Chrome, Mozilla Firefox, Opera, Microsoft Edge and Apple Safari as candidates for browser-based password managers. The selection was based on the user count for cloud-based and on the assumption that most users come across with browser-based password managers as they are built-in.

The analysis should create an overview of operating system and browser support, a table of free and paid features, default settings, description of security for authentication (e.g., hash functions), and data security (e.g., encryption and default settings) for cloud-based password managers. And upsides and downsides for browser-based password managers.

During the time of testing[1], new updates of the selected password managers and browsers were introduced regularly, therefore the support of the operating system, available features, default settings, security, user count, and pricing may be different in the future.

4.1 Cloud-based Password Managers

Cloud password managers typically provide a browser extension, a standalone app, and a mobile application. Extensions are usually limited in most cases since their focus is on capturing and filling passwords in web browsers. Therefore, standalone desktop applications are also provided for better vault management. The main feature of cloud-based password managers is the ease of synchronisation across all devices, therefore, they should be available on all major desktop and mobile operating systems. The possibility of multiplatform support comes with complications, as some operating systems or browsers may require the usage of native APIs, e.g., to ensure security on mobile devices. Most of them have a free version for personal use, but also provide paid plans with extra features. They also focus on enterprise customers with business plans which provide further customisation and management options. This will also present additional information in the description

[1] From 09/2020 to 05/2022

of each chosen password manager.

The following table shows the support across different browsers[2]:

Browser extensions						
Password Manager	**Google Chrome**	**Mozilla Firefox**	**Opera**	**Microsoft Edge**	**Safari**	**Tor**
LastPass	yes	yes	yes	yes	yes	no
Dashlane	yes	yes	no	yes	yes	no
1Password	yes	yes	no	yes	yes	no
Roboform	yes	yes	no	yes	yes	no
Sticky Password	yes	yes	yes	yes	yes	no
Bitwarden	yes	yes	yes	yes	yes	yes

The following table shows the support across Windows, MacOS, Linux, iOS, and Android operating systems:

Native desktop and mobile applications					
Password Manager	**Windows**	**MacOS**	**Linux**	**iOS**	**Android**
LastPass	no	yes	no	yes	yes
Dashlane	yes	yes	no	yes	yes
1Password	yes	yes	yes	yes	yes
Roboform	yes	yes	yes	yes	yes
Sticky Password	yes	yes	no	yes	yes
Bitwarden	yes	yes	yes	yes	yes

4.1.1 Features

Password managers in this category share several main features: multiplatform support and multidevice synchronisation, autofill with the browser extension, password generator. Aditionally, they do not only store passwords, but also provide a solution to store credit card numbers and other documents, notes, or files in secure storage, TOTP (Time-based One-time Password), and more. Many cloud-based password managers provide a "password health-check" (also called Vault Health Reports), informing the user how secure the password is or if the account was found in some leaked database, which went public. Some of them support multifactor authentication (MFA), but for advanced MFA such as hardware authentication, a paid version is needed in many cases.

[2]The table shows the support according to password managers providers, even though in most cases the extension support is dependent on used web browser's engine.

The following two tables compare the availability of the main features of each password manager:

Features						
Password Manager	**Autofill**	**Password generator**	**Device sync**	**Password sharing**	**Breach check**	**Health check**
LastPass	free	free	free	free	paid	paid
Dashlane	free	free	paid	free	paid	paid
1Password	paid	paid	paid	paid	paid	paid
Roboform	free	free	paid	paid	none	free
Sticky Password	free	free	paid	paid	none	free
Bitwarden	free	free	free	paid	none	paid

Features						
Password Manager	**File storage**	**Software MFA**	**Hardware MFA**	**Other items (e.g., IDs)**	**Custom fields**	**TOTP**
LastPass	paid (1 GB)	free	paid	free	free	paid[3]
Dashlane	none	free	paid	free	none	none
1Password	paid (1|5 GB)	paid	paid	paid	paid	paid
Roboform	none	free	none	free	none	none
Sticky Password	none	free	none	free	none	none
Bitwarden	paid	free	paid	free	free	paid

All tested password managers provide a password generator option. General customisation (lowercase, uppercase, numbers, symbol options) is mostly the same across all password managers, but they vary in a few characteristics such as minimum and maximum length and special character set. The length is not crucial. On the other hand, a special character set is more critical since different sites may enforce different password policies with special symbols, therefore the customisation of the special character set is highly appreciated.

[3]Feature available only for business plans

4.1.2 Security

Cloud password managers generally use similar security features. The following table shows the database encryption standard and the key-derivation function for authentication.

Security		
Password Manager	**Database Encryption**	**Key Derivation Function**
LastPass[4]	AES-256	PBKDF2-SHA-256
Dashlane[5]	AES-256	Argon2 or PBKDF2-SHA2
1Password[6]	AES-256	PBKDF2-HMAC-SHA256
Roboform[7]	AES-256	PBKDF2-SHA-256
StickyPassword[8]	AES-256	PBKDF2
Bitwarden[9]	AES-256	PBKDF2-SHA-256

4.1.3 Candidates of Cloud-based Password Managers

4.1.3.1 LastPass[10]

LastPass is used by 25.6 million users and 70 000 companies. For better usability, LastPass offers "How It Works" a site that provides a simplified guide on how to download, create, account, add an existing user password, generate, and save a new password. Interestingly, a master password does not require a special character, making it easier to remember, but less secure.

Pricing of LastPass:

- Personal
  - Free
  - Premium: $3.00/month
  - Families (max. 6 users): $4.00/month
- Business
  - Teams: $4.00/user/month
  - Business: $6.00/user/month

The free plan limits multidevice synchronisation to one device type (computer or mobile) and does not provide password health checks, which can be considered a downside.

[4] http://www.lastpass.com/enterprise/security
[5] http://www.dashlane.com/download/Dashlane_SecurityWhitePaper_November2020.pdf
[6] http://support.1password.com/1password-security/
[7] https://help.roboform.com/hc/en-us/articles/115003926191-RoboForm-for-Business-Security-Overview
[8] https://www.stickypassword.com/downloads/Sticky_Password_Security_WhitePaper.pdf?v=3
[9] https://bitwarden.com/images/resources/security-white-paper-download.pdf
[10] https://www.lastpass.com

4.1.3.2 Dashlane[11]

Dashlane is used by more than 14 million users and 18 000 companies. Dashlane offers "How It Works" a site with standard password manager information and a simple how-to-use guide.

Pricing of Dashlane:

- Personal
  - Free
  - Premium: $6.49/month
  - Families (max. 6 users): $8.99/month
- Business
  - Teams: $5.00/user/month
  - Business: $8.00/user/month

The free option limits the storage of max 50 passwords and 1 device. Premium and Family option offers unlimited storage and devices, plus a VPN in the plan.

4.1.3.3 1Password[12]

1Password is used by more than 15 million users and 75 000 companies. 1Password opens a video on how to use a guide after the first login into the extension.

Pricing of 1Password:

- Personal
  - Standard Account: $2.99/month
  - Family Account (max. 5 users): $4.99/month
- Business
  - Teams (max. 10 users): $19.95/month
  - Business: $7.99/user/month

1Password does not offer a free plan, therefore it may not be appealing for first-time password manager users.

4.1.3.4 RoboForm[13]

RoboForm opens the site with a detailed online tutorial.

RoboForm pricing:

- Personal
  - Free
  - Premium: $23.88/year
  - Family (max. 5 users): $47.75/year
- Business
  - RoboForm for Business: $39.95/user/year

[11] https://www.dashlane.com
[12] https://1password.com
[13] https://www.roboform.com

Free plan does not provide device synchronisation and two-factor authentication. A list of other features is available on RoboForm's website.

4.1.3.5 Sticky Password[14]

Sticky Password is used by more than 2 million users.

Sticky Password pricing:

- Personal
  - Free
  - Premium: $29.99/year or $199.99/lifetime
- Teams
  - Premium for Teams: $29.99/user/year

Key feature of Sticky Password is the possibility of using the password manager on USB devices with Windows OS as a local password manager, which may be appreciated by security-orientated users. A major downside is that the free account does not include device synchronisation, which may be a limitation for many users. A list of other features is available on the Sticky Password's website.

4.1.3.6 Bitwarden[15]

Bitwarden is an open-sourced password manager, used by millions of individuals and businesses.

On its website, they provide a "Getting Started" page explaining the steps to create a user account and a detailed description on how to use its products. The basic steps are self-explanatory and similar across all other password managers, but users may also find a guide for the self-hosting option if they choose so.

Bitwarden pricing:

- Personal
  - Individual
    - Basic Free Account
    - Premium Account: $1/month
  - Sharing
    - Free 2-Person Organization
    - Family Organisation (max. 6 users): $3.33/month
- Business
  - Teams Organisation: $3/user/month
  - Enterprise Organisation: $5/user/month

Basic Free Account is sufficient for the average user, but certain features which can be considered crucial are available only in Premium account, e.g., a Vault Health Reports, and hardware Multifactor Authentication (YubiKey, etc.). Since this thesis focusses mainly on password managers used by the average user, a complete table of available features for each plan, including the enterprise, can be found on Bitwarden's website and will not be discussed in this topic.

[14] https://www.stickypassword.com
[15] https://bitwarden.com/

4.2 Browser-based Password Managers

Every browser offers an internal password manager. For end-users, this is usually the first password manager with which they come in hand, since all popular browsers have the built-in password manager enabled by default. This does increase the overall password managers' users, but it does not increase the security or improve the user's security behaviour. On the other hand, a browser-based password manager is quite limited compared to the features provided by cloud password managers. As the main focus is on managing passwords typed on websites, users may face an issue if they want to store credentials that are not bound to any website, or to store additional information for the login process, such as answers to security questions. Browser-based password managers work without the need for an account, but the user can synchronise and backup their passwords when they log in with an account specific to the browser.

Security aspects of Browser-based Password Managers are well described in a paper "That Was Then, This Is Now" by Sean Oesch and Scott Ruoti[17]. In their paper, a "Table 6: Overview of Password Vault Encryption" shows that passwords are stored in .sqlite database, or in .json format. System browsers (Edge and Internet Explorer for Windows, Safari for MacOS) rely on the operating system to handle password storage by using native options: Windows Vault for Windows, MacOS Keychain for MacOS. Encryption is provided by the operating system in most cases. Only Firefox handles encryption on its own with 3DES and uses SHA-1 as a KDF function.

Passwords stored with browsers' password management are commonly targeted by malware, e.g., a recent report by AhnLab ASEC[16] shows that the autofill option was misused to steal user's stored credentials.

4.2.1 Candidates of Browser-based Password Manager

4.2.1.1 Google Chrome

Google Chrome password manager can be analysed in two stages, with and without a Google account.

- With Google account and synchronisation on: the behaviour is similar, with the exception that the user has the option to generate a password by using the "suggest password" option. All passwords are then synchronised within the Google account.
- Without Google account: the browser automatically offers the user to save a password when filling out a registration or website log-in form. In browser settings, the user can manage saved passwords and is also warned if it is using a weak password or the account has been compromised.

4.2.1.2 Mozilla Firefox

Mozilla Firefox browser offers a password generator without the need for an account. Mozilla company also offers an iOS and Android application to synchronise passwords between the Firefox browser and the mobile device using Mozilla's account. Firefox offers "Firefox

[16] https://asec.ahnlab.com/en/29885/

Monitor" a feature to users with an account, which checks the user's emails against a list of known security breaches and informs the user if their email has been comprised.

4.2.1.3 Opera

Opera browser offers a password generator without an account. One downside is that password synchronisation is limited only to the Opera browser using an account.

4.2.1.4 Microsoft Edge

Without an account, Microsoft Edge offers to save passwords, but does not provide a password generation with is enabled, when the user is logged into a Microsoft account. Microsoft Edge lacks a health-check feature.

4.2.1.5 Apple Safari

Safari browser is available only on MacOS and iOS. The password manager is provided by a system application called Keychain. When the user is registering on a new site, a suggested stored password is offered to the user. All passwords can be seen in the Safari settings window, protected by a computer's login password. If the password for a certain login item is weak or reused, the user is presented with a warning.

4.3 Conclusion

Password managers from cloud category do not have big differences, most of them use the same security and support most browsers and operating systems. Differences occur in the offered advanced features and in the source code availability and monetisation. The question arises whether unpaid password managers are less secure, but according to the table of used security mechanisms, this is not true. I just would like to emphasise that some password managers support hardware key security, which can be considered an important security feature. The key advantage of password managers in a browser is that they can be used without an account, but with various restrictions. Using a given browser account allows additional features to get closer to cloud category password managers. Overall, browsers' password managers may be enough for the average user.

Chapter 5

Pre-implementation Testing

Password managers' user base is still growing, but its features and design are changing with each version[6]. To find out how average user setups and uses a password manager, a direct observation test is concluded, the results will be used in determining an ideal password manager and later in the implementation of a custom password manager.

5.1 Direct Observation Test

The direct observation test was conducted with a person, who has no professional technical skills and does not use an external password manager, but is a daily Internet user. The chosen password manager was a LastPass extension (version 4.62.0) for the Google Chrome browser. The participant will be provided with the following tasks:

- Create a LastPass account
- Download an extension
- Login into LastPass extension
- Create an account on a test site and save newly created credentials into LastPass (the password is chosen by the tested user)
- Login to the site by using an autofill feature
- Change password to a generated one from a password manager
- Express feelings about using a password manager

The goal of this test was to actively observe the possible issues that could occur when setting up a password manager. The findings are summarised in the following chapter.

5.2 Results of Direct Observation Test

5.2.1 Direct Observation Test Procedure

The participant had difficulty with orientation on the LastPass site, but proceeded without any bigger problems after a *Get LastPass Free* button was noticed. Interestingly, the test user did not visit the "How it works" page. Creating an account was straightforward as the registration form followed standard requirements. To my concern, the Master Password requirements are quite low with only the following conditions being required:

- 12 characters long

- At least one number
- At least one lowercase letter
- At least one uppercase letter

I would expect a special character requirement to enhance master password security. A useful hint was provided when the user was actively notified when the corresponding requirement was met. After creating an account, the user was redirected to the Chrome web store, where the LastPass extension was offered to download. Adding the extension followed the standard procedure. Interestingly, the user had been automatically logged in with the credentials created on the LastPass site, which means that LastPass was keeping an active session, checking if the extension was added, and placing a lower burden on the user.

Creating an account on a test site, the test was carried out without any problems, where LastPass automatically offered to generate a new password while filling in the registration form. The password manager feature was available by clicking on a dark grey icon, which can be considered indistinct. The tester missed the password auto-generator function on the first attempt, as no active indication was provided. After submitting the form, the LastPass extension actively offered a dialogue suggesting adding credentials to the password manager's vault. During the login procedure, the user eventually noticed the dark grey icon feature, which automatically filled in the stored credentials. After changing the password, the LastPass extension offered to update the password as expected.

5.2.2 User's Self Evaluation

The test user felt unsure during orientation on the LastPass site, which could have been caused by not being a native English speaker, but has expressed a positive feeling about the important options (in this case, *Get LastPass Free*) being highlighted. After adding the extension, the user was unsure what to expect, but this could have been caused by skipping the usage example page. The user was generally concerned with downloading an unknown extension. However, the user was quite satisfied and would be willing to continue using the extension even after finishing the test.

5.2.3 Conclusion

In summary, the user was able to register and download a password manager. LastPass provides a *How It Works* section on its website, but the user was straightforward in creating the account and no usage tutorial was offered after signing up, which could prevent the initial confusion of features provided by the password manager. By my understanding, the user would not look for an external password manager without a known person's recommendation.

Based on the observed behaviour, I would add the requirement "tutorial" that the user must complete before using the password manager. To improve overall security, a special character would be appreciated as a required character when creating the account and choosing the master password.

Chapter 6

Ideal Password Manager

Describing the ideal password manager is not an easy task. Considering that not everyone is technically based, it is important that the ideal password manager is easy to use for all users. In addition to storing passwords, the password manager should increase the overall security of user behaviour. Here, we come to the problem where security stands against usability. Therefore, it is necessary to find a balance between security and usability requirements when implementing a password manager. This can also be achieved with customisation options. Some security features, such as automatic logout, may be left for the user to modify or to disable, but it is preferred to enable these security features by default and warn the user of possible consequences and security downgrades when turned off.

6.1 Features

The ideal password manager client application should run on all commonly used platforms. In the best case, the features described in the chapter on basic functionality should be available. Basic features include the easy addition of a new password and further viewing, searching, editing, and deleting of existing passwords. In addition, it should provide a tutorial on how to use the password manager. The user should be able to synchronise their passwords, as today's typical user uses multiple devices. Another necessary feature should be a password generator with the ability to modify required parameters such as length, lowercase and uppercase letters, numbers, and special characters. Since most users use passwords mainly on websites, it is required that the ideal password manager offers the extension for the most popular browsers to make it easier to enter passwords on websites.

6.2 Usability

Users avoid trying new things, especially older users, who fear being scammed, especially with login credentials with which they can access systems containing private information. Therefore, it is important to properly explain how the password manager works and what benefits it brings. From a usability point of view, a password manager must minimise the burden on its users. Users mainly use login credentials on web browsers. Therefore, a browser extension should automatically prompt the user to save, input, and generate new passwords when creating a new account. UI elements can help the user to remind that the password manager is installed. The user must feel better when saving the credentials to the

password manager than when writing the email and password down on a piece of paper. The user also needs to synchronise all passwords across all devices they use, regardless of the operating system or browser.

6.3 Security

When the passwords are stored on the central server, the master password must be properly hashed, so no one can view or crack the original password. Therefore, the use of appropriate hash functions such as SHA-256 and other techniques such as salting is important. The entire user's vault needs to be encrypted with an advanced cryptic algorithm, such as AES with a key of at least 128 bits long, which is derived from the master password.

Other security features must be implemented, such as memory scrubbing, keylogger detection, and automatic lock or logout.

Chapter 7

Implementation of Password Manager

This chapter describes an implementation of a custom password manager. Previous chapters describe the best security practices for programming passwords storage, and the overall security features of password managers, including UX elements that provide additional security (e.g., automatic logout, etc.). Based on the acquired knowledge, I identified the important functions in "ideal password manager" section. This gave me the necessary knowledge and resources to design and implement my own multiplatform password manager.

7.1 Problem

As in any other project, we need to define the problem to which we want to provide solution. Nowadays, users need to remember a lot of passwords and, at the same time, each password should be secure (in terms of used characters, length, uniqueness) to ensure better security of the corresponding account, as shown in figure: 7.1. It is not possible to remember all the passwords which may result in lowering the password strength, or add up to passwords reuse, which both decreases the account's security. The user also needs to be aware of possible dangers on the Internet, such as phishing attacks, etc.

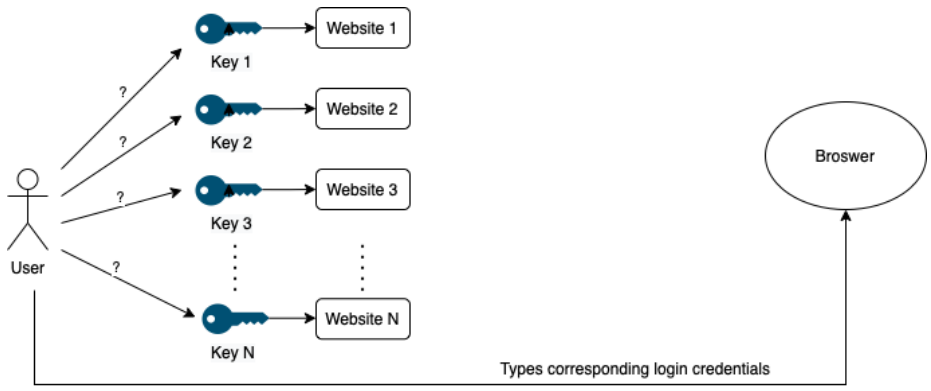

Figure 7.1: Scheme of the problem: Remembering all passwords

7.2 Solution

The solution to the problem is to provide a secure storage for all user's passwords, as shown in figure: 7.2. Possibly with options to generate a new secure and unique password and to fill the stored passwords on to the corresponding websites. The user needs to remember only one password, under which all other login credentials will be encrypted. It is also required to provide vault synchronization across all the devices that the user has.

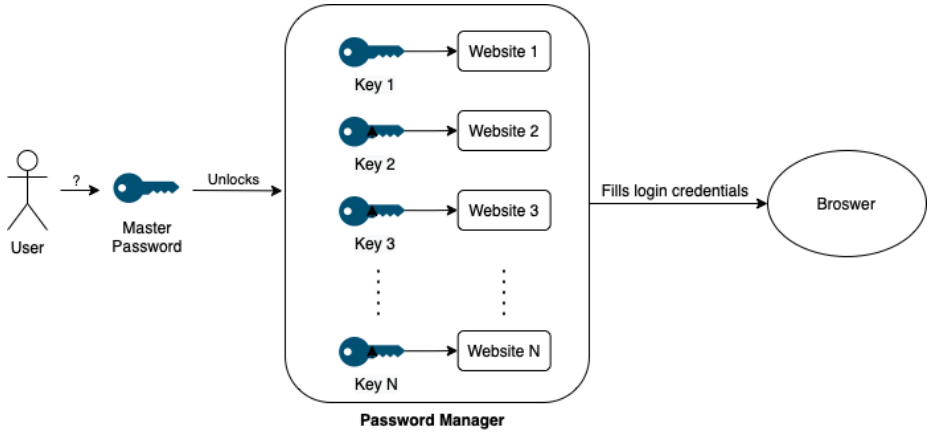

Figure 7.2: Scheme of the solution: Using password manager

7.3 Implementation

The project includes desktop application, browser extension and server. Desktop application will offer key features described in the previous sections. The application will run on Linux, MacOS, and Windows operating systems. The extension will be available for browsers running on the Chromium browser engine. Server will be available as a Docker container.

7.3.1 Application

To easily create a cross-platform (Linux, Windows, MacOS) password manager, I decided to use the Electron[1] framework, which is an open-source software framework for creating cross-platform applications from GitHub. It offers development tools for building applications across operating systems and allows user interface development using HTML, CSS, and JavaScript. To create the front-end (UI) part of the application, I used the React[2] framework, which makes it easier to create user interfaces (using HTML, CSS, and JS) and is compatible with the Electron framework. The application can use node packages, which helped me to create necessary features.

[1] https://www.electronjs.org/
[2] https://reactjs.org/

Important packages list:

- @material-ui[3]: provides Material-UI elements
- axios[4]: provides HTTP client functions
- electron-store[5]: provides persistent data storage for Electron applications
- crypto[6]: node library providing cryptographic functions
- js-crypto-hkdf[7]: external library providing HKDF function
- sqlite3[8]: provides native connector for sqlite3 database
- ws[9]: provides WebSocket server functions

7.3.2 Extension

The extension in Chromium[10] browsers acts like a web page. I used the React[11] framework with @material-ui package to provide equal UI as the electron application has. The Chrome API[12] is used to fill in the login details on the website. The API provides access to information about the currently opened website in the browser and options to manipulate with the page. The extension depends on the application, the user must have the application installed to be able to use it. The extension communicates with the application, which passes and processes the password records. Communication takes place via a WebSocket server.

7.3.3 Server

The server is implemented as a node application. To make it easier to set up and turn on the server, a docker-compose[13] file is provided to create a Docker container, which sets up a required MySQL database and the node server that works with the database. The scheme of the communication is introduced in figure: 7.3. The server provides an API for registration, login, and manipulation of stored user's records.

The implemented Docker container provides the user option to easily run and modify the server (e.g.,: PORT, database credentials, etc.) on their own device. The option is available for more technically proficient users who care about the increased security of their data stored in the password manager.

Important packages list:

- sequelize[14]: provides ORM (Object-Relational Mapping) for MySQL and other databases.
- express[15]: provides web framework functions

[3] https://mui.com/
[4] https://github.com/axios/axios
[5] https://github.com/sindresorhus/electron-store
[6] https://nodejs.org/api/crypto.html
[7] https://github.com/junkurihara/jscu
[8] https://github.com/TryGhost/node-sqlite3
[9] https://github.com/websockets/ws
[10] https://www.chromium.org/
[11] https://reactjs.org/
[12] https://developer.chrome.com/docs/extensions/reference/
[13] https://docs.docker.com/compose/
[14] https://github.com/sequelize/sequelize
[15] https://github.com/expressjs/express

- mysql2[16]: provides MySQL drivers

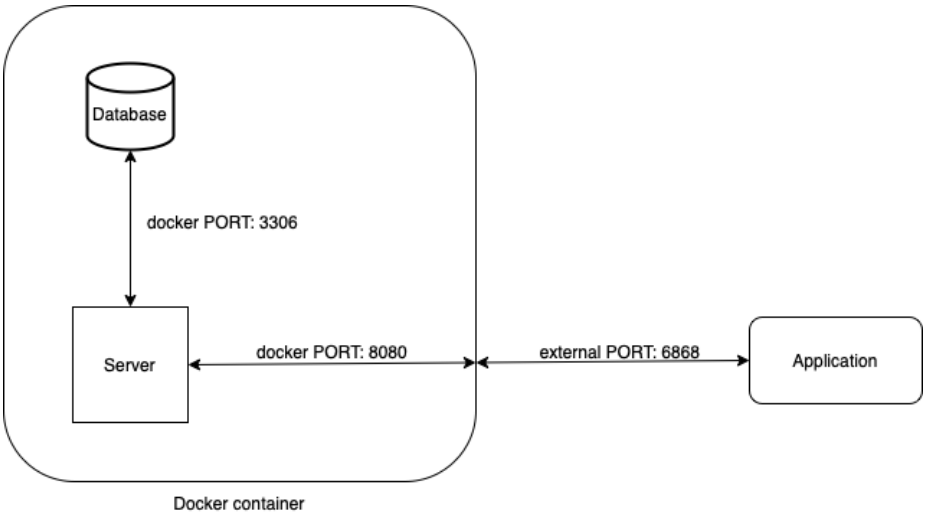

Figure 7.3: Communication among parts of the project

The user must first register in the registration window in the application, this will create security keys for database encryption and authorization. Decryption and encryption is possible only in the application. The server stores only the user's name, email, and hash passwords used to verify the login.

7.3.4 Communication

The figure: 7.4 shows a simplified communication between the individual parts of the project and interactions by the user. The user can register, log in and manage passwords in the application. User browses the web pages and fills in the passwords for the current website with a help of the extension, which also allows passwords management. The browser and the extension communicate using the Chrome API. The application communicates with the extension using a WebSocket server. Communication between the server and the application is provided via HTTPS requests (with SSL security). The server communicates with the database using a sequilize library.

[16] https://github.com/sidorares/node-mysql2

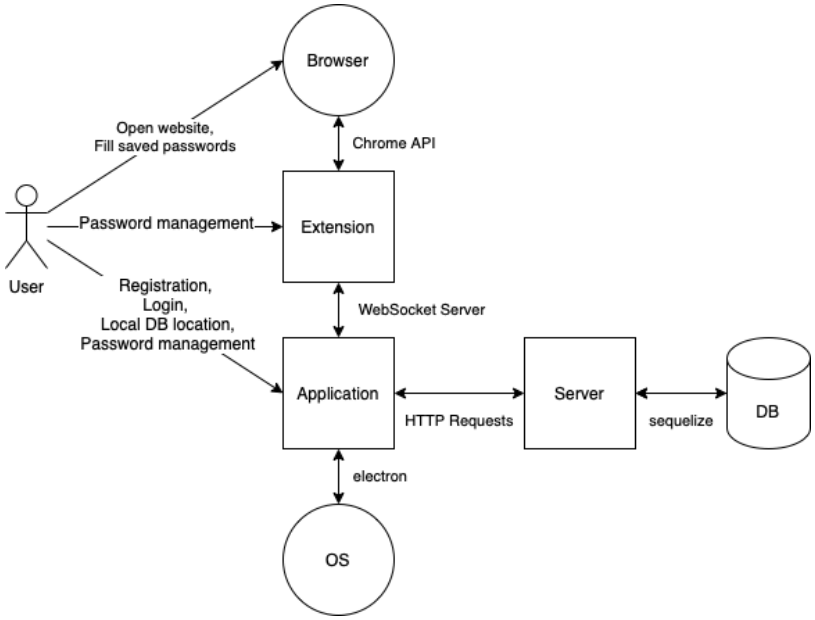

Figure 7.4: Communication among parts of the project

7.3.5 Developement

For system versioning, I used a GitHub repository. Source codes for all project parts are available on following URLs:

- Application: https://github.com/hader00/password-manager-dev
- Extension: https://github.com/hader00/password-manager-extension-dev
- Server: https://github.com/hader00/password-manager-app-server

7.4 Required Security

Security needs to be considered both from an authentication perspective and from a secure data storage.

Authentication is the process of verifying that the authenticated person is the person whom he or she claims to be. This process is verified using the login name, in our case, email address and password. The password is the most critical point, as it is used to obtain the key to decrypt the database. Using the hash function, the password can be stored on the server so that the server owner cannot obtain the password in cleartext form.

The user who registers, chooses an email address and a password for the account. The scheme of the registration process is illustrated in figure: 7.5. The password is normalised in the case that the user chooses other characters than ASCII. A master password is generated from the password in the client part using the PBKDF2-SHA512 function with 100,000 iterations. This hash is sent along with the email to the server, where the PBKDF2-SHA512 function with 100,000 iterations will be used again with a custom salt. The email and the resulting hash are stored in the database, and the user is assigned an identification number

under which the individual user data will be stored. Additionally, a symmetric key and an initialisation vector (IV) are generated with a random generator function on the client side. These keys are then used to decrypt the database. An encryption key is generated from the password and email using HKDF-SHA512. The symmetric key is then encrypted with the client's encryption key, and the encrypted symmetric key with IV is also sent to the server.

Login has a similar security process as registration. The scheme is illustrated in figure: 7.6. the user enters an email and password, a master password hash is created in the client part, and this information is sent to the server, which retrieves the hash and compares it with the already stored login credential in the server if it finds valid, the user's database, encrypted symmetric key, and IV are sent back to the client application. Encryption and decryption occur only on the client's part, thus preventing the misuse of data stored on the server.

Both figures: 7.5 and 7.6, were inspired by documents Security Whitepaper[2] from Bitwarden[17] and Technical Whitepaper[1] from LastPass[18].

[17] https://bitwarden.com/
[18] https://www.lastpass.com/

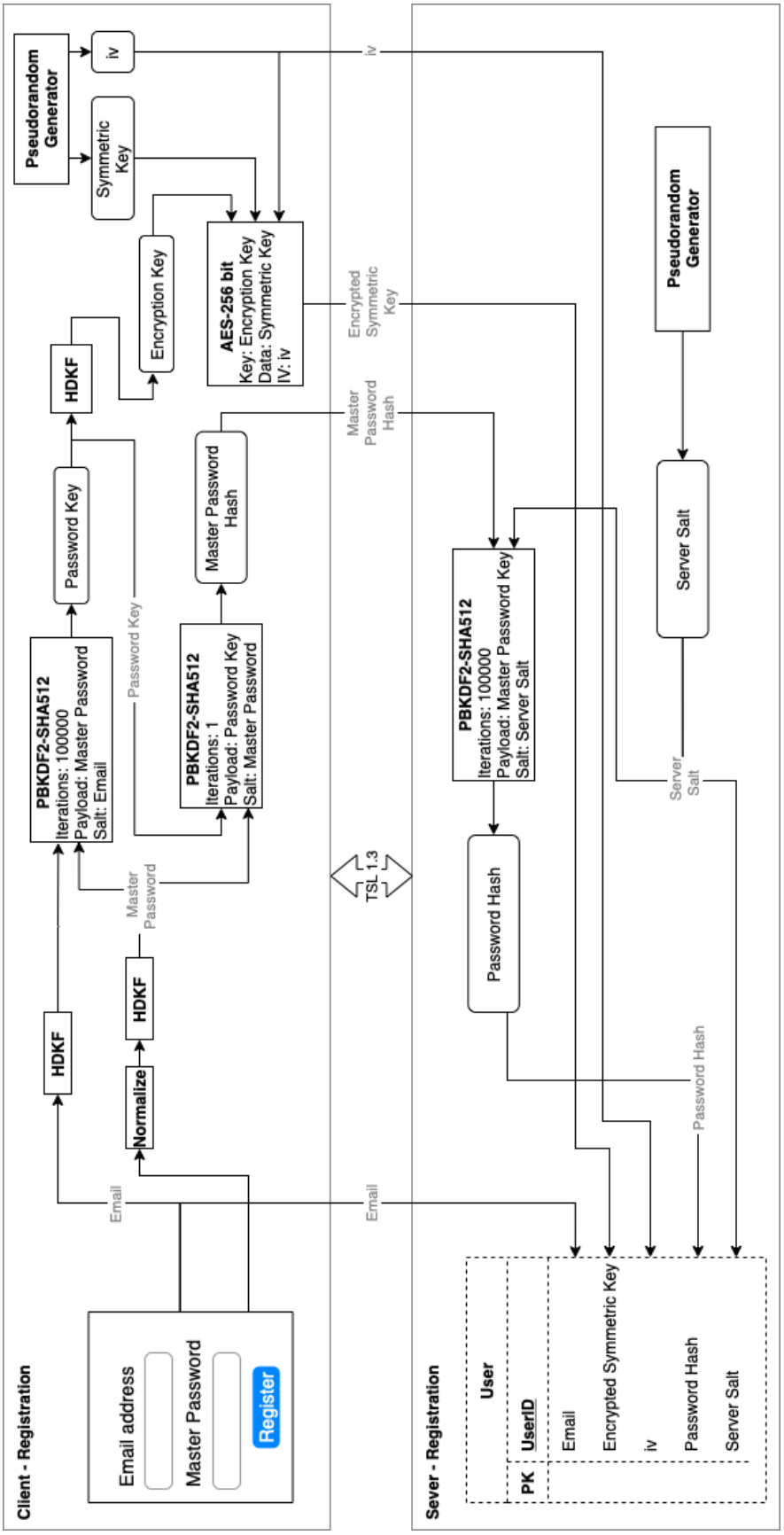

Figure 7.5: Registration process

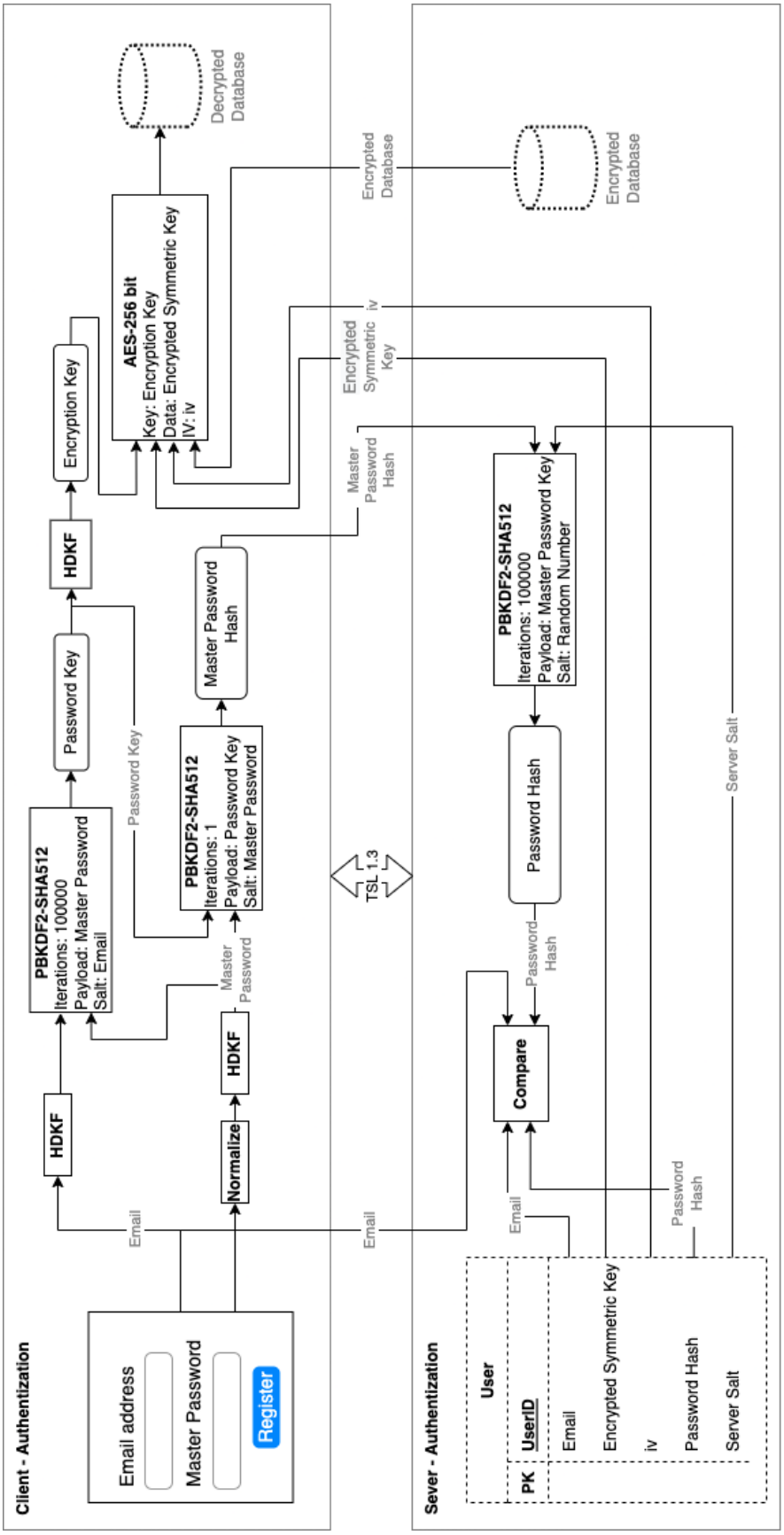

Figure 7.6: Authentication process

7.5 UI/UX

The user interface (UI) and user experience (UX) play a big role in application usage. The application consists of individual views that offer the options to manipulate stored passwords. The arrangement of the individual views is designed, so that the user can navigate in the application as easily as possible. Registration (figure: 7.8) and Login (figure: 7.7) view are standard, with fields for name, surname, email, and the master password. After a successful registration or login, the user is presented with a list of password items (figure: 7.9), which includes a field for search and option for viewing, and editing of individual passwords. View providing options for adding new passwords (figure: 7.10) is accessible using the "+" button or keyboard shortcut. In password editing view, user can find a password generator (figure: 7.11) option. The application provides account view (figure: 7.12), which is accessible by the application's top bar menu. The extension also provides similar views as the application, with an additional field in the passwords view, where it presents a password item attached to the website the user is currently on for easier and faster password filling.

The application can be used in two modes:

- Local: the password database is stored locally on the computer in .sqlite format
- Remote: the password database is stored on a remote server, which can be:
  - Default: the password database is on the default server, running at: `https://password-manager-mysql.herokuapp.com`
  - Custom: the password database is stored on a user-created Docker server that can run locally on a PC or on other server

7.5.1 Local mode

To use the local database, the user must create the database first, by clicking *LOCAL REGISTRATION* button to display local registration window. The password needs to fulfil required parameters. After clicking on *CREATE DATABASE* button the database will be created in default *APP_DATA* directory, according to operating system:

- Linux: ~/.config/Simple Password Manager
- MacOS: ~/Library/Application Support/Simple Password Manager
- Windows: C:\Users\%USER_NAME%\AppData\Roaming\Simple Password Manager

The user can choose a different path where to store the database by using *Custom Location* switch and selecting the path in the dialog that appears.
After a successful creation of a local database, a window with an empty password list will appear.

7.5.2 Remote mode

7.5.2.1 Default server

The default server runs at `https://password-manager-mysql.herokuapp.com`. The database will be stored on this server if the user registers an account using the *CREATE ACCOUNT*

button. In the registration window, user needs to fill required information and by clicking *REGISTER* button, the registration process will start. The application will automatically log in the user on successful registration and display windows with an empty list of passwords.

7.5.2.2 Custom Docker server

The Docker server must be started according to the instructions in the repository. Default server port is 6868, but the user can change it. The registration is similar as in the previous section, only during registration the user must enter the server address using the *Custom Server* switch. The default address is: `http://localhost:6868`. The same server must be specified when logging in.

7.5.3 Application features

7.5.3.1 Add password record

Using the + button in the passwords list window, or by using keyboard shortcut Ctrl + N (Windows, Linux) or CMD + N (MacOS) will open a window allowing the user to add a new password. The password record can contain the following elements:

- Title - password record name (required field)
- Username - username or email password entry
- Password - user password
- Description - description of the password record
- URL - the address of the password record

After filling the fields and clicking on *SAVE* button, the password record will be saved and added to the list of password records.

7.5.3.2 View password record

In case, the user wants only to view a password. The password view window can be opened using the *VIEW* button for each password in the list window. The opened window does not allow editing the record, only displaying and copying.

7.5.3.3 Edit password entry

The password editing window can be opened by using the *EDIT* button by each password in the password list window, or by using *Pencil* icon in the password view window. The opened window allows editing individual fields of the password, option to save changes and to delete the password.

7.5.3.4 Delete password entry

The password entry can be deleted by using *Trash* icon in the password edit window, or by using the keyboard shortcut Ctrl + Backspace (Windows, Linux) or CMD + Backspace (MacOS). After clicking on the button or using the keyboard shortcut, a pop-up window will appear, where the user must confirm the deletion.

7.5.3.5 Open password's website

To open password's website in the default browser, the user can use *VISIT PAGE* button. This feature is useful when the user is viewing passwords in the application and not in the extension.

7.5.3.6 Generate password

It is possible to generate a random password in editing password windows. The user can choose password requirements to include special characters, numbers, lowercase letters, uppercase letters, and password length. Clicking on the *GENERATE PASSWORD* button will generate a new password accordingly to the selected requirements.

7.5.3.7 Copy password

Using the *Copy* icon in the password view window or in the password editing window, the password will be copied to the system clipboard. The password will be deleted from the clipboard after a certain time, more information on automatic deletion can be found in the following paragraph.

7.5.4 Advanced application features

7.5.4.1 Account management

In the top panel of the application, the user can find a *User* section and the *Account Settings* option, which will open an account management window. Here, the user can choose the time after which the application will be automatically log out. And the time after which the password copied from the application or from the extension will be removed from clipboard. These settings are application-specific. The default time for automatic logout is 5 minutes, the default time for automatic clipboard deletion is 10 seconds. The user can also find *Clear app data* option in the *User* section of the panel, which will clear all data stored for the application.

7.5.4.2 Automatic logout

After logging in, the countdown will start (the time is given by the value *Vault Timeout*, which can be found in the user account window). The countdown will reset when the user is actively using the application, specifically when adding, viewing, or editing a password record. Logging off the main application also logs off the user from the extension.

7.5.4.3 Automatic clipboard deletion

After copying the password using the copy buttons, the countdown will start (the time is given by the value *Clear Clipboard*, which can be found in the user account window). To delete a password from the clipboard, the application needs to be active and in the foreground, in case of inactivity, the password from clipboard will be deleted as soon as the application or extension is in the foreground again. This shortcoming is due to the limitations of the technologies used for development.

7.5.4.4 Export and import password records

In the top panel of the application, user can find *Items* section and two options: *Export Passwords* and *Import Passwords*. The export option prompts the user for input credentials and allows him to select a folder to store the exported database, which will have the name: *%YEAR_MONTH_DAY_HOUR_MIN_SEC%.tsv*. This file can be imported (for example to a second account) using the import database button.

7.5.5 Extension features

The extension offers the same features as the application. It offers some extra features to make it easier to work with the website.

7.5.5.1 Fill in the password record

The user can find a *FILL CREDENTIALS* button in the password view window, which when pressed, fills in the data of the displayed password on the currently open web page.

7.5.5.2 Fill for the currently open web page

In the passwords list window, user can find another list called *Current website*. This list displays the records for the currently open web page with the *FILL* button, which fills in the saved login details, when the button is clicked. This feature speeds up your password manager.

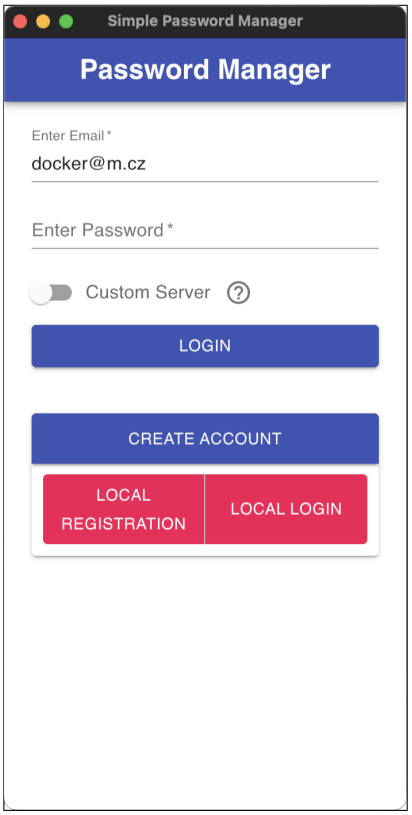

Figure 7.7: Login view

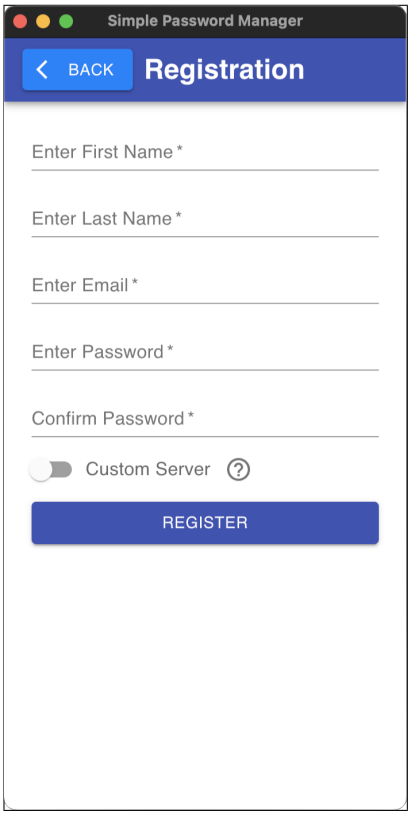

Figure 7.8: Registration view

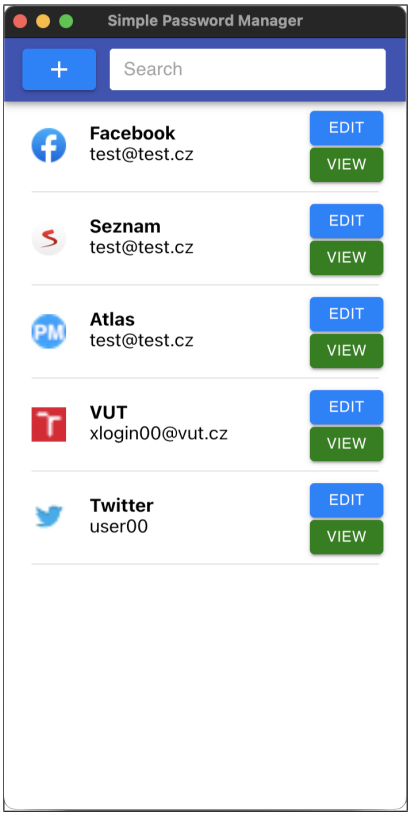

Figure 7.9: Password list view

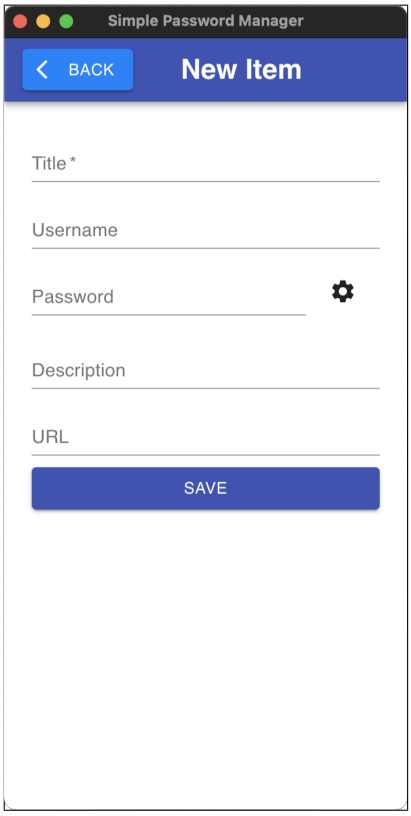

Figure 7.10: Add new password

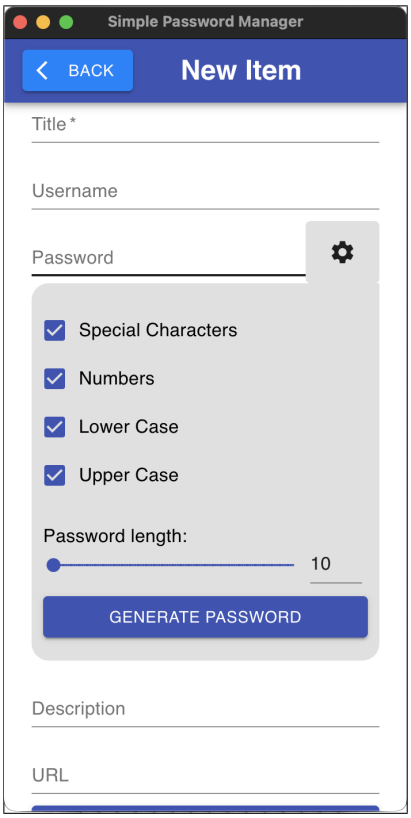

Figure 7.11: Password generator

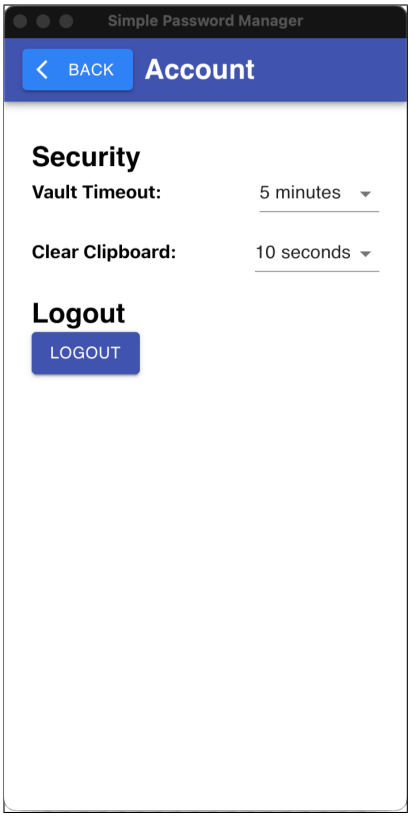

Figure 7.12: Account view

7.6 Testing

The goal of testing is to determine whether the implemented functions are sufficient and meet the expectations of potential users. The application was continuously tested during development on multiple operating systems, by my friends and colleagues.

The testing proceeded as follows: the user has installed the application and browser extension. When the new version was released, the user installed them again.

During the first test, the user is introduced to the functions and to account creation. The testers will work with those accounts during further tests. The aim of the first test was to find out whether the registration and login is sufficiently clear. There were no complications.

The second test is to add credentials to services that the user uses regularly. Because some testers already use other password managers, they would prefer to import existing vaults from other password managers. However, this functionality is complicated to implement because different password managers have their own schemes for password export. Therefore, a function would have to be created for each existing password manager to properly import passwords. This feature could be added at a later stage of the application development. In any case, users had no problem adding new items to the tested password manager.

The third test was about the browser add-on. The add-on requires that the main application be turned on. This requirement was not understood by some testers, but if the user chooses to store passwords locally, it is technically necessary for the add-on to communicate with an application that works with the local database. In this test, a tester recommended an option for the user to fill in the password more easily if he or she is on a page for which there are login details. I took this into account and added the feature by introducing another field in the password list view.

The fourth test focused on generating new passwords. Complications, in the form of UI bugs, occurred during the first version of the application and have been resolved in the next version of the application. Here, it was recommended to implement further advanced settings, where the user could choose the minimum and maximum of each category (special characters, numbers, lowercase and uppercase letters).

During the tests, few bugs were found that have been corrected. The testers appreciated the simple user interface and the ease of use of the application. Those who already use a password manager would appreciate an add-on that would run on other browsers and a mobile application for easy synchronisation.

7.7 Limitations

The implemented password manager meets the basic requirements in terms of security and usability. This is not a full-fledged password manager. Other features such as password strength check, database check, passwords import, and other items could be implemented in next versions of the password manager. From the point of view of usability, it would be appropriate to implement add-ons for other popular browsers, such as Mozilla Firefox and Apple Safari. It would be convenient to develop mobile applications for systems such as Android and iOS.

The application, add-on, and server offer basic password management features. The result of the functionality was confirmed by the users who participated in the testing during development. However, the application cannot be considered full-fledged. Today, the user needs the password manager to work on mobile devices and across different browsers.

From a usability point of view, the user is limited to storing login details containing an email and a password. Currently, existing password managers offer the possibility to securely store other records, e.g., payment card data, identification card data, etc.). The add-on only works for Chromium browsers, which currently have the largest share of [19] on the market, but if the user uses a different browser, he or she cannot take full advantage of this project.

From a security perspective, it would be a good idea to have a server with your own SSL certificates. In addition, there is a need for more secure communication between the add-on and the application, which currently runs via WebSocket [20]. It is also advisable to think about the general security of node projects, when imported packages may contain errors, it would be more appropriate to write the application natively to allow better memory management to prevent possible data leaks from the application.

[19] https://gs.statcounter.com/browser-market-share
[20] https://javascript.info/websocket

Chapter 8

Conclusion

Password manager is software that allows to securely store not only login credentials, but also other sensitive data, therefore it is important to analyse this topic. This thesis focuses on different categories of password managers and their security.

The work is divided into two parts. In the first part, the aim was to describe passwords from the perspective of authentication, the role of password managers, and to introduce security-related components not only of password managers, but of the overall secure storage and manipulation of data. Emphasis was placed on the provided features, categorisation by the used environment, their advantages, and disadvantages, possible security risks, and usability shortcomings. A list of common features and categories are presented in this part.

In the second part, which was more practical, an experiment was conducted to reveal how people use password managers. On the basis of the information from the first part and the results of the experiment, an ideal password manager was described that offers the most fundamental functionality.

In the last part, an implementation of an ideal password manager was offered, in which it was tested whether potential users could use the password manager without any problems and whether it would make it easier for them to log into the websites. On the basis of the results, it can be concluded that even a simple password manager is sufficient for users and increases the efficiency and security of the user on the Internet.

This work is intended to help readers make easier decisions when choosing a password manager and to outline the security mechanisms that password managers use. Password manager developers can find a description of the features that may be beneficial to users.

In further work, a deeper look at the security that password managers use, such as database types and how data are handled, can be evaluated.

Bibliography

[1] *Technical Whitepaper*. LastPass, Jul 2019. Available at: https://support.lastpass.com/download/lastpass-technical-whitepaper.

[2] *Bitwarden Security Whitepaper*. Bitwarden, Inc., Oct 2020. Available at: https://bitwarden.com/images/resources/security-white-paper-download.pdf.

[3] Barker, E. *Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms*. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, march 2020. DOI: 10.6028/NIST.SP.800-175Br1. Accessed on 14.03.2022. Available at: https://doi.org/10.6028/NIST.SP.800-175Br1.

[4] Cheng, L., Liu, F. and Yao, D. D. Enterprise data breach: causes, challenges, prevention, and future directions. *WIREs Data Mining and Knowledge Discovery*. 2017, vol. 7, no. 5, p. e1211. DOI: https://doi.org/10.1002/widm.1211. Accessed on 14.03.2022. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1211.

[5] Fagan, M., Albayram, Y., Khan, M. and Buck, R. An investigation into users' considerations towards using password managers. *Human-centric Computing and Information Sciences*. Berlin/Heidelberg: Springer Berlin Heidelberg. 2017, vol. 7, no. 1. DOI: 10.1186/s13673-017-0093-6. ISSN 2192-1962. Accessed on 14.03.2022. Available at: https://doi.org/10.1186/s13673-017-0093-6.

[6] Flourentzos, H. A Usability Evaluation and Re-design of the password manager software KeePass2. Master of Science School of Informatics University of Edinburgh. 2018. Available at: https://groups.inf.ed.ac.uk/tulips/projects/1718/keepass2.pdf.

[7] Grassi, P., Fenton, J., Newton, E., Perlner, R., Regenscheid, A. et al. *Digital Identity Guidelines: Authentication and Lifecycle Management [includes updates as of 03-02- 2020]*. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, march 2020. DOI: 10.6028/NIST.SP.800-63b. Accessed on 14.03.2022. Available at: https://doi.org/10.6028/NIST.SP.800-63b.

[8] Gray, J., Franqueira, V. N. L. and Yu, Y. Forensically-Sound Analysis of Security Risks of Using Local Password Managers. In: *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*. IEEE, 2016. ISBN 9781509036943. Accessed on 14.03.2022.

[9] Hoonakker, P., Bornoe, N. and Carayon, P. Password Authentication from a Human Factors Perspective: Results of a Survey among End-Users. *Proceedings of*

the Human Factors and Ergonomics Society Annual Meeting. Human Factors and Ergonomics Society. 2009, vol. 53, no. 6, p. 459–463. ISSN 1071-1813. Accessed on 14.03.2022.

[10] Independent Security Evaluators, I. *Password Managers: Password Managers' Secrets Management | ISE*. 2019. Accessed on 14.03.2022. Available at: https://www.ise.io/casestudies/password-manager-hacking/.

[11] *Information technology — Security techniques — Information security management system — Requirements*. Standard. Geneva, CH: International Organization for Standardization, october 2005. Accessed on 14.03.2022.

[12] Krawczyk, D. H. and Eronen, P. *HMAC-based Extract-and-Expand Key Derivation Function (HKDF)* [RFC 5869]. RFC Editor, may 2010. DOI: 10.17487/RFC5869. Accessed on 14.03.2022. Available at: https://www.rfc-editor.org/info/rfc5869.

[13] Luevanos, C., Elizarraras, J., Hirschi, K. and Yeh, J.-H. Analysis on the Security and Use of Password Managers. In: IEEE. *2017 18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. IEEE, 2017. ISBN 9781538631515. Accessed on 14.03.2022.

[14] McMillan, R. *The World's First Computer Password? It Was Useless Too*. 2012. Accessed on 04.12.2021. Available at: https://www.wired.com/2012/01/computer-password/.

[15] Meltem Sönmez Turan, W. B. and Chen, L. *Recommendation for Password-Based Key Derivation*. Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, december 2010. DOI: 10.6028/NIST.SP.800-132. Accessed on 14.03.2022. Available at: https://doi.org/10.6028/NIST.SP.800-132.

[16] Nurse, J., Creese, S., Goldsmith, M. and Lamberts, K. Guidelines for usable cybersecurity: Past and present. In: *2011 Third International Workshop on Cyberspace Safety and Security (CSS)*. October 2011, p. 21 – 26. DOI: 10.1109/CSS.2011.6058566. Accessed on 04.12.2021. Available at: https://doi.org/10.1109/CSS.2011.6058566.

[17] Oesch, S. and Ruoti, S. That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Thirteen Password Managers. *CoRR*. 2019, abs/1908.03296. Accessed on 14.03.2022. Available at: http://arxiv.org/abs/1908.03296.

[18] Qadir, A. M. and Varol, N. A Review Paper on Cryptography. In: IEEE. *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. 2019, p. 1–6. DOI: 10.1109/ISDFS.2019.8757514. Accessed on 14.03.2022. Available at: https://doi.org/10.1109/ISDFS.2019.8757514.

[19] Raza, M., Iqbal, M., Sharif, M. and Haider, W. A survey of password attacks and comparative analysis on methods for secure authentication. *World Applied Sciences Journal*. Citeseer. 2012, vol. 19, no. 4.

[20] Rescorla, E. *The Transport Layer Security (TLS) Protocol Version 1.3* [RFC 8446]. RFC Editor, august 2018. DOI: 10.17487/RFC8446. Accessed on 14.03.2022. Available at: https://www.rfc-editor.org/info/rfc8446.

[21] Sasse, M. A., Steves, M., Krol, K. and Chisnell, D. The great authentication fatigue–and how to overcome it. In: Springer. *International Conference on Cross-Cultural Design*. 2014, p. 228–239. Accessed on 4.12.2022.

[22] Sasse, M. A. Usability and trust in information systems. In:. Edward Elgar, 2005.

[23] Singh, M. and Garg, D. Choosing Best Hashing Strategies and Hash Functions. In: IEEE. *2009 IEEE International Advance Computing Conference*. 2009. DOI: 10.1109/IADCC.2009.4808979. Accessed on 14.03.2022. Available at: https://doi.org/10.1109/IADCC.2009.4808979.

[24] Standards, N. I. of and Technology. *FIPS 197, Advanced Encryption Standard (AES)*. National Institute of Standards and Technology, november 2001. Accessed on 14.03.2022. Available at: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[25] Stobert, E., Safaie, T., Molyneaux, H., Mannan, M. and Youssef, A. ByPass: Reconsidering the Usability of Password Managers. Gina Cody School of Engineering and Computer Science. 2020. Accessed on 14.03.2022. Available at: https://users.encs.concordia.ca/~mmannan/publications/Bypass-SecureComm2020.pdf.