A Password Manager with Focus on Security and Usability

Tony Shu, Yuxuan Zeng, Maolin He

Contribution by each member

Yuxuan Zeng(1358012): Design the password generation algorithm. Design the User Interface of the password manager. For the coding part, implement all the functions that are expected, including 1. image importing, cutting, and mapping. 2. implement all action functions connect to the mouse activities listener: selecting tiles, btn clicking, window jump, etc. 3. implement all calculation logic. 4. Data storage, reading, encryption, and decryption logic. For the report, contribute to User Interface and Workflow part.

Tony Shu(1305712): Design a strategy for recovering the generated passwords when only parameters are stored. Explore insightful evaluation metrics to provide a reasonable estimate of the strength of the generated password. Analyze the security of the proposed password manager via various aspects. Report: Complete the entire section of Method, including Multiple Master Code, Master Code Representation, Password Generation, Data Storage, and Error Tolerance for Legitimate Users. Complete the entire section of Evaluation, including Evaluation Metrics, Generated Password Strength, Master Code Security, Data Storage Security, and Usability.

Maolin He(135008): Analyzed how to apply the measures proposed in the study to the password manager(PM) we developed. In particular, this includes recognizing the legitimate user by matching incorrect input to the previous incorrect input made by the same legitimate user, as well as ensuring password security by storing only the public parameters in the password manager database without master codes or passwords' raw data. Report: 1. completed the entire section of the introduction, including a brief description of the dilemmas of passwords, the trade-off between the security and usability of password managers, and the research problem. 2. completed the entire section of the literature review, including the literature on password manager design and the papers evaluating password managers' usability and/or security. 3. completed the entire section of the conclusion, including a restatement of the study's aim, a summary of the research, and a recommendation for future work.

1. Introduction

Passwords play a crucial role in protecting most of our valuable information and data, so security should be a central issue for passwords. However, complex and random passwords are rarely used. One possible reason is that secure passwords may not be usable because human memorization is limited. Security issues are often left behind as the secondary concern in the trade-off between the strength and memorability of passwords. Users prefer some particular passwords that are relatively easy to remember, even if such passwords are usually weak. Vulnerable passwords are generally straightforward and likely to have patterns or relate to personal information, such as name and date of birth, which is very effortless for attackers to guess or crack. In support of solving the problem above, a wide variety of password managers have been developed. Some particular software focuses on security, while others concentrate on usability. However, how to design a password manager that maximizes both security and usability remains an open question.

The trade-off between usability and security should be considered when designing a password manager:

1. People are likely to enter the wrong password for various reasons, such as forgetting the correct password or accidentally entering the wrong password. So for usability, a password manager should have error tolerance for legitimate users. However, incorrect passwords may be due to brute force attacks. So, for security, it also needs to have a countermeasure for entering incorrect passwords.

2. Only having one master code for the entire password manager is undoubtedly the most convenient option. However, it faces the security risk that all passwords will be exposed once the master code has been cracked. In comparison, it would be the most secure and inconvenient that a master code can only safeguard a password.

3. Master code is not secure if it is simple, but master code is not easy to remember if it is complex.

Based on the previously mentioned trade-off between the usability and security of password managers, this study aims to answer the following question: how can the security and usability of password managers be guaranteed at the same time when the trade-off between these two is faced?

2. Literature review

Stobert and Biddle designed Versipass, a password manager that does not store passwords but only image cues of graphical passwords. It facilitates secure password reuse by allowing users to categorize passwords and use the same image cue for multiple passwords that belong to the same category [9]. However, users may feel that they lose control of their passwords because passwords are invisible to users. Versipass allows users to select grid blocks as an image cue, which may make Versipass vulnerable to hotspot attacks. Hotspot attacks take advantage of the fact that specific grid blocks in an image are more often selected as part of the password.

Almuhanna et al. designed a system that uses honey encryption as a countermeasure to brute-force attacks. The system recognizes brute-force attacks by matching the entered hash master password with auto-generated honeywords. If there is no match, the users can try two more times [2]. Although using honey encryption for an image cue is challenging to implement, the idea of giving the user more chances after identifying the user can be applied.

There are some factors associated with the usability of a password manager. For example, Karole et al. suggest that users are reluctant to lose password control, which means that giving users password control should be considered in the password manager design to improve usability [8]. Another relevant factor is password strength feedback, Adams and Sasse suggest that this feedback on password strength during password generation is essential [1]. Obviously, the feedback can make users more aware of the benefits of generated passwords. Frykholm and Juels propose [6] that error tolerance for legitimate users is also one of the critical factors of usability, which can improve the user experience. Another vital usability factor is the memorability of the master code. A user-friendly password manager should consider the user's cognitive ability. Table 1 is a brief summary of factors that can improve the usability of password manager:

Usability
Users have password control
Feedback on password strength
Error tolerance for legitimate users
Memorability of the master code

Table 1. The factors of Usability

There are some factors that can improve the security of a password manager. Ciampa et al. correctly argued that users worry that all passwords would expose once attackers crack the master code [4], since the protection of the master codes is one of the most crucial security factors of a password manager. According to Camp's research, a secure password manager needs to be able to lessen the effects of any attack [3]. In other words, countermeasures for the attacks that could happen in the future should be one of the most important security factors of a password manager. Another essential security factor of a password manager is

the strength and randomness of the generated passwords. The users generally hope that if an attacker wants to crack a text-based password generated by the password manager, the attacker would need a lot of resources. At least, the cost of cracking the password must be greater than the benefit that obtaining the password can bring. Secure password storage is also a significant security factor of a password manager. It should be mandatory to encrypt the stored data using strong encryption before storage. According to the research of Zhao et al., some password managers choose not to store the metadata of clients' master codes and other stored passwords [10].

Security
The protection of the master codes
Countermeasures for the attacks
The randomness and Strength of generated password
Safe password storage

Table 2. The factors of Security

3. Method

This section mainly focuses on the five ingredients of the proposed password manager to address the research question. Moreover, it outlines why those components are crucial for a secure and usable password manager.

3.1. Multiple Master Code

A master code consists of multiple concatenated characters from the input alphabet, which is a required input whenever the user attempts to access any stored data regarding one specific account in the password manager. Master code security is one of the most vital features of a password manager since it is utilized as the primary authentication verification method in many mainstream password managers. However, if an attacker compromises the master code itself, all passwords stored in the password manager are exposed, leading to a significant loss of confidentiality. Based on this security threat, a secure password manager should have the functionality to allow multiple master codes. If a targeted attack compromises one master code, passwords guarded by the rest of the master code are still secure unless further attacks are mounted. Although this strategy does not necessarily prevent such attacks, it increases the computational overhead to crack all stored passwords, as only a subset of them is exposed with the same cost.

3.2. Master Code Representation

Similar to the requirements of a secure password, a master code should also be complex for better security. Additionally, as discussed earlier, allowing multiple master codes is crucial in ensuring security. Nevertheless, the argument raises the following contradiction: If users are capable of memorizing many complex master codes, password managers will become meaningless since users can remember their raw passwords instead. Thus, utilizing a master code representation that is relatively easier to memorize than traditional textual master code plays a vital role in the proposed password manager. According to the research conducted by Stobert and Biddle, graphical passwords can be considered a promising alternative to textual passwords since they function better in memorability, even though which exact form of graphical passwords performs better is to be determined [9]. Hence, the master code representation can be derived from their research findings, which employ graphics as the primary representation.

3.3. Password Generation

Password managers' general responsibility is to automatically generate strong passwords for users within an acceptable amount of time. Thus, cryptographic hashing functions can be considered advantageous candidates for the task as they furnish the following features: efficiency, one-way property, pseudo-randomness, second preimage-resistant, and collision-resistant. In an analytical study, Gupta and Kumar clearly described the main difference between the Message-Digest 5(MD5) and Secure Hash Algorithm 256-bit (SHA-256), which are commonly deployed in reality. Their research suggests that despite the higher complexity SHA-256 demands, SHA-256 has superior security than MD5, measured in various aspects [7]. Therefore, the presented password manager will utilize SHA-256 as the primary algorithm for generating passwords to seek better security, though with the cost of slightly lower efficiency.

Additionally, the inputs to the password generation algorithm can be formally described as follows:

$$password = SHA256(URL||username||mastercode) \tag{1}$$

where the concatenation of URL and $username$ represents a clear destination where the password is used for authentication, and $mastercode$ is utilized to ensure only the legitimate user can generate the password, based on the assumption that the master code is known to the user only.

3.4. Data Storage

A typical threat in password security is that the attacker may gain access to the data storage. All confidential information will thereby be in the clear if passwords are stored directly. In response to such a security threat, a reasonable

password manager should never directly store any secret information in the data storage medium. Hence, the proposed password manager only stores three pieces of information for each password it generates, including the website URL, the username, and the hash value of the corresponding master code that safeguards the password. The URL and username should not be kept as secret information as they are meant to be publicly accessible. However, in contrast, the confidentiality of the master code is remarkably crucial. Therefore, it should be stored securely such that the attacker cannot gain any useful information if the data storage is compromised. One that can be utilized to achieve this goal is to store the hash value of the master code produced by SHA-256, based on the assumption that the preimage and second preimage resistant holds.

During the authentication phase, if the user enters a master code whose hash value matches the stored hash value, it is deemed that the user is genuine. The manager can then re-calculate the password if required since all parameter needed for generating the password is accessible.

3.5. Error Tolerance for Legitimate Users

Many password managers will automatically be locked to improve security if the number of incorrect master codes entered exceeds a certain threshold. Nevertheless, due to the lack of human memorability, even the genuine user can sometimes unintentionally make mistakes when entering the master code. Therefore, a password manager must recognize legitimate users and provide more allowed attempts to them in order to enhance usability. This requirement indicates the significance of distinguishing between a genuine user and an attacker when it is observed that a wrong master code is provided. A straightforward strategy to achieve this task is that the password manager treats any user who enters the correct master code within a few attempts as legitimate. Then, the password manager records each mistake the user has made(if any). Based on the assumption that the user is likely make the same error again, the password manager can allow more attempts if any of the recorded mistakes are observed during the authentication process.

4. User Interface

The login interface of the password manager is mainly responsible for setting up new categories, entering and verifying the master code. Figure 1 shows the interface when users set up a new category with the randomly selected image tiles as master code. Figure 2 is the screenshot when the users choose an existing category and are ready to verify the master code.

The Get Password interface (Figure 4) is used to generate new passwords for users and to display recorded account password information, it will automatically pop up once the user is authenticated.

The interface shown in Figure 5 will pop up once users click the 'Check Strength' button, the outcome of this window is generated by the $zxcvbn$ algorithm, which is discussed in section 7.1.1.

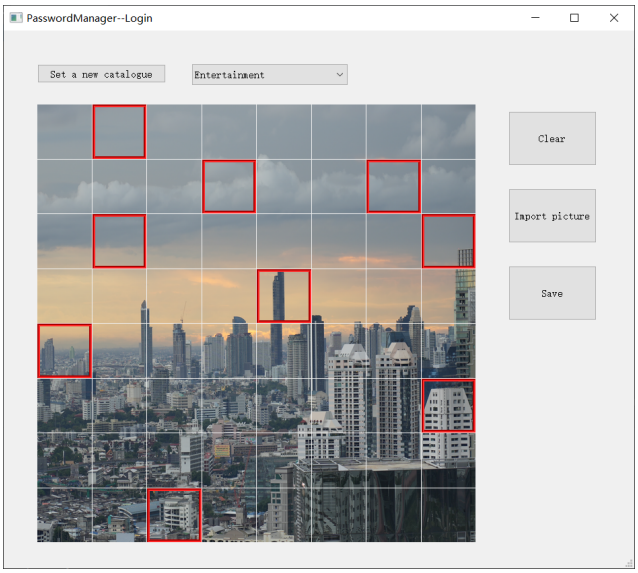

Figure 1. Interface when creating a new category

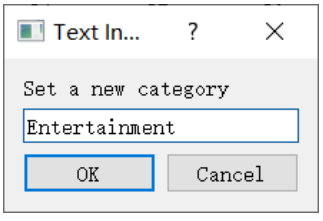

Figure 2. Login page's pop-up when setting a new category

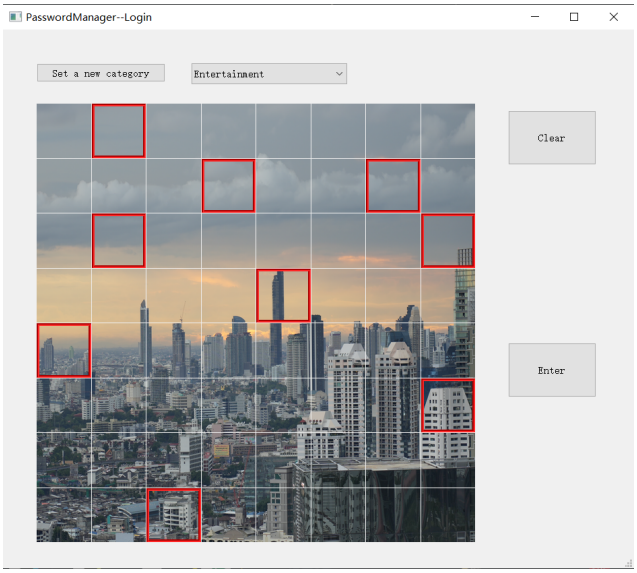

Figure 3. Interface when choosing an existing category to log in

Figure 5. Password Strength Interface

5. Workflow

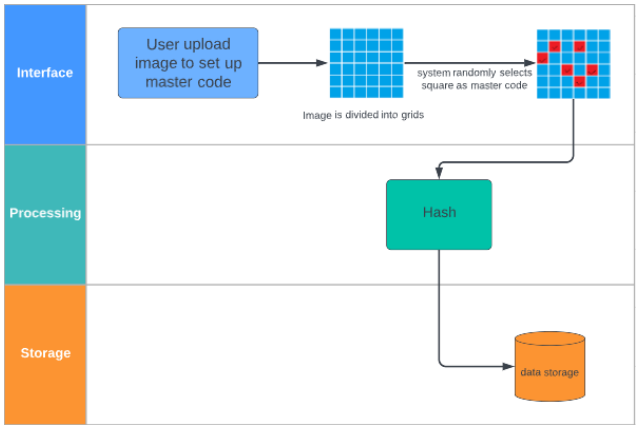

Figure 6. Save new category and its graphic password

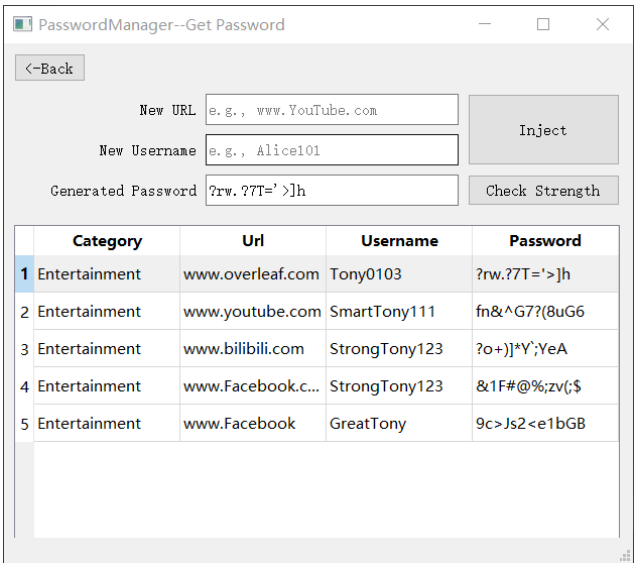

Figure 4. Get Password page of the Password Manager

Figure 6 demonstrates the workflow of saving new categories and their graphic master code. To create a new password category, the user clicks the ”Set a new category” button and enters a category name into the pop-up window. Once the user clicks the ”save” button, the hash value of the coordinates of the selected image tiles and the imported image are stored in the data storage.

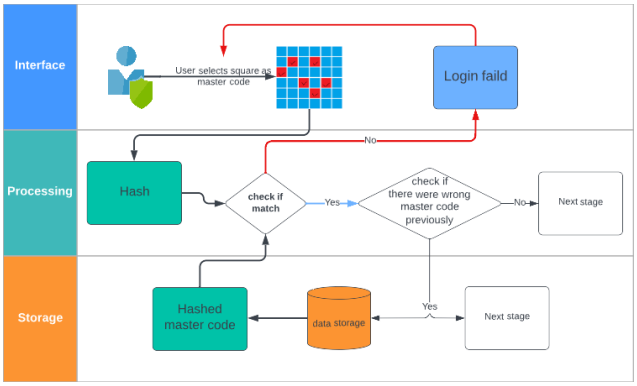

Figure 7. Graphic Master code verified workflow

Figure 7 demonstrates the workflow of the master code verification. To access the Get Password page of the existing category, the users are required to select the correct tiles. Once users get authenticated, the next stage window will automatically pop up. The user will fail to access the next stage if the hash value of the master code that the user enters can not match that stored in the data storage. The user has three chances to get authenticated by entering the correct master code, and the category will be locked if fail to do so. The error tolerance mechanism is working all the time, it will record the last 1 to 2 wrong attempts when the user finally gets verified and gives the user more attempts if the recorded wrong attempts are observed again.

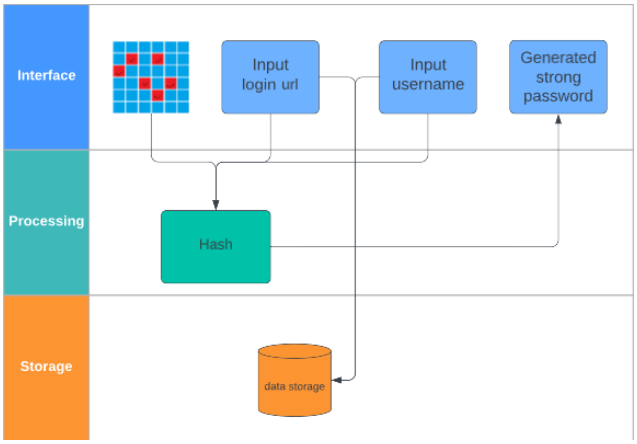

Figure 8. Password generation workflow

Figure 8 illustrates the workflow of password generation. The user interface shown in Figure 4 will automatically pop up once the user passes the verification process and the master code that the user entered will be temporarily held in the memory cache as one of the password-generating parameters, URL and username are the other two parameters required to generate the password. The system will then simply concatenate these three together and SHA-256 hash function will be applied to generate the password. The URL and username will be stored in the data storage while the process finalized.

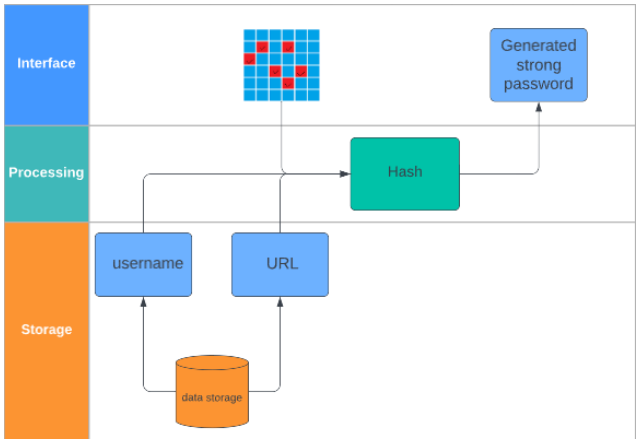

Figure 9. Password generation workflow

Figure 9 demonstrates the process of re-calculation of the password at runtime. The system extracts usernames and URLs from the data storage, calculates with the master code to get the password then displays it on the table list as Figure 4 shown.

6. Results

URL	Username	Password	#Guesses required
www.overleaf.com	Tony0103	?rw.?7T='>]h	10^{12}
www.youtube.com	SmartTony111	fn&^G7?(8uG6	10^{12}
www.bilibili.com	StrongTony123	?o+)]*Y`;YeA	10^{12}
www.Facebook.com	StrongTony123	&1F#@%;zv(;$	10^{12}
www.Facebook.com	GreatTony	3A9$.xAjD$0P	10^{12}

Table 3. Password Strength table

7. Evaluation

7.1. Security

7.1.1 Evaluation Metrics

With modern advanced attack models, it has become paramount to construct a systematic approach to measure password strength properly. Wheeler's research proposed a sophisticated evaluation algorithm named $zxcvbn$, which involves machine learning techniques, minimum ranking over frequent-used password lists, and pattern matching. The research also indicates that this emerging algorithm can

accurately estimate the strength of passwords under real-world online attacks, although with an overestimation of strength for offline attacks. The $zxcvbn$ algorithm guides the following evaluation of generated passwords by the proposed password manager.

7.1.2 Generated Password Strength

The research conducted by Florencio et al. clearly described two typical attack models for cracking passwords - online and offline attacks. Online attacks run on the targeted server's public interface, and the server itself returns the authentication outcome. In contrast, offline attacks run and are tested on the attacker's own hardware. Furthermore, their research illustrates the security lower bound for the two attack models measured in the number of guesses required - 10^6 for online attacks and 10^{14} for offline attacks [5]. Table 3 illustrates the number of guesses needed to crack some generated passwords, estimated by the $zxcvbn$ algorithm as mentioned earlier. Although all generated passwords are likely to survive online attacks by exceeding the corresponding threshold, still, they are not strong enough to confidently endure offline attacks. Because none of the generated passwords achieved a higher number of guesses required than the security lower bound for offline attacks. Hence, a more sophisticated password generation algorithm should be involved to enhance the strength of generated passwords, such as salted hashing and iteratively executing the hash function multiple times instead of only executing once.

7.1.3 Master Code Security

Since the master code is chosen by the password manager instead of relying on the user's habituation, it may increase the difficulty of constructing dictionary attacks and reduce the risk of suffering from such attacks. Moreover, only a subset of passwords is exposed if an attack is successfully mounted against one particular master code, provided that the users are willing to take advantage of the functionality of multiple master codes and adequately manage them. However, unfortunately, this particular master code representation is vulnerable to shoulder surfing, meaning that the master code itself can be directly observed if the user enters the master code in front of the public or the screen is monitored by malicious software. The above vulnerability can be reduced using several authentication techniques, such as two-factor authentication and biometrics. However, since each has its benefits and drawbacks, no single technique is entirely secure against modern attacks. Combining multiple techniques to authenticate users may be a promising solution, but it undoubtedly negatively impacts user experience.

7.1.4 Data Storage Security

Based on the assumption that the one-way property and the second preimage-resistant of SHA-256 holds, the possibility of the stored master code's confidentiality being violated remains relatively low. Nevertheless, the current method of storing data does not have any particular defense strategy against data modification, leading to a potentially severe violation of availability. For example, if an attacker modified any of the stored hash values of master codes, even the legitimate user cannot pass the authentication verification.

7.2. Usability

The proposed password manager provides some functionalities to improve usability. For example, all generated passwords are visible and managed by the legitimate user, resulting in the user feeling in control of their passwords. In addition, users are informed regarding the strength of each generated password, which allows users to gain more knowledge related to the security level of their passwords. Moreover, the proposed password manager supports error tolerance for the legitimate user, reducing the negative impact of unintentionally entering an incorrect master code. Furthermore, the master code representation is in a graphical form, requiring lower memorability than traditional textual passwords.

However, since this password manager is merely a prototype, further improvements can be made to enhance usability. One that can be considered is to support the password manager across multiple devices. Because users generally own multiple devices, it is indeed problematic if passwords are stored within only one specific device. Another advancement in usability is implementing the functionality of password auto-fill-in since manually typing in the generated passwords is trivial.

8. Conclusions and Future Directions

This study aims to research how a password manager can ensure security and usability simultaneously. Therefore, in this study, we develop a password manager that considers both security and usability in the following ways: applying multiple graphical master codes into the password manager to reduce the risk of all passwords being completely compromised while maintaining memorability. In addition, the proposed password manager stores only the parameters needed for generating passwords and re-calculates them at runtime to avoid violating confidentiality. Moreover, it provides error tolerance to seek a better user experience based on the assumption that the user is likely to enter the same incorrect master code again during the verification phase. Our evaluation shows that although our password manager is imperfect and improvements are still required, it provides insight to enhance usability and security simultaneously.

Future work is to improve the usability and security of the password manager further. In terms of usability, our password manager should be modified to support multiple platforms, such as Android, IOS, and Windows. It should also support cloud storage so it can be used across devices. Additionally, the password manager should be able to recover and reset passwords and master codes. Regarding security, more advanced hashing functions can be embedded into the password manager to generate stronger passwords, such as iteratively executed salted hashing.

References

[1] Anne Adams and Martina Angela Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[2] Albatoul AlMuhanna, Afnan AlFaadhel, and Anees Ara. Enhanced system for securing password manager using honey encryption. In *2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU)*, pages 150–154. IEEE, 2022.

[3] L Jean Camp. Design for trust, trust, reputation and security: Theories and practice. ed. rino falcone, 2003.

[4] Mark Ciampa, Mark Revels, and John Enamait. Online versus local password management applications: An analysis of user training and reactions. *Journal of Applied Security Research*, 6(4):449–466, 2011.

[5] Dinei Florêncio, Cormac Herley, and Paul C Van Oorschot. An administrator's guide to internet password research. In *28th large installation system administration conference (LISA14)*, pages 44–61, 2014.

[6] Niklas Frykholm and Ari Juels. Error-tolerant password recovery. In *Proceedings of the 8th ACM conference on Computer and Communications Security*, pages 1–9, 2001.

[7] Piyush Gupta and Sandeep Kumar. A comparative analysis of sha and md5 algorithm. *architecture*, 1(1):5, 2014.

[8] Ambarish Karole, Nitesh Saxena, and Nicolas Christin. A comparative usability evaluation of traditional password managers. In *International Conference on Information Security and Cryptology*, pages 233–251. Springer, 2010.

[9] Elizabeth Stobert and Robert Biddle. Memory retrieval and graphical passwords. In *Proceedings of the ninth symposium on usable privacy and security*, pages 1–14, 2013.

[10] Rui Zhao, Chuan Yue, and Kun Sun. Vulnerability and risk analysis of two commercial browser and cloud based password managers. *ASE Science Journal*, 1(4):1–15, 2013.