**Logo, company name

Description automatically generated**

Software Requirements Specification & Designs Document

on

**Safe Vault**

**[A Password Managing Tool]**

*(Submitted to* ***Jain (Deemed-to-be-University), Bengaluru*** *as a part of Project Centric Learning**for the partial fulfillment of the degree of* ***Bachelor of Computer Application)***

**CTIS/ISMA - Vth Semester**

**Session 2021-22**

*Submitted by*

**Jasbir Singh** [20BCAR0111] **CTIS**

**Suman Garai** [20BCAR0246] **ISMA**

**Trishul V Biradar** [20BCAR0146] **CTIS**

**Umang Basantani** [20BCAR0247] **CTIS**

**Utsav** [20BCAR0265] **CTIS**

*Guided by*

**Dr. Taskeen Zaidi**

**Assoc. Professor, Department of BCA,**

**Jain (Deemed-to-be) University, Bangalore.**

****

**CERTIFICATE**

Certified that the project work entitled Password Managing Tool has been carried out by **Suman Garai (20BCAR0246), Utsav (20BCAR0265), Umang Basantani (20BCAR0247), Trishul V Biradar (20BCAR0164) and Jasbir Singh (20BCAR0111)** who are bonafide students of Jain University, in partial fulfillment for the award of Bachelor of Computer Applications in ‘CLOUD TECHNOLOGY AND INFORMATION SECURITY’ AND ‘INFORMATION SECURITY AND MOBILE APPLICATION’, School of Computer Science and IT, Jain (Deemed-to-be University), Bangalore, during the year 2021-22.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

|  |  |
| --- | --- |
| **Faculty Guide Signature:**  **Name: Dr. *Taskeen Zaidi***  **Designation: *Assoc. Professor*** | **HOD**  **Signature:**  **Name: *Dr. Suchitra Nair*** |

|  |  |
| --- | --- |
| **Name of the External Examiner 1.**  **2.** | **Signature with Date** |

**DECLARATION**

We hereby declare that the project work entitled “**PASSWORD MANAGING TOOL**” submitted to the Jain (Deemed-to-be University), Bangalore during the academic year 2021-22, is a record of an original work done by me/us under the guidance of **Dr.** **Taskeen Zaidi**, ***ASSOC. PROFESSOR***, Department of BCA, School of CS & IT, Jain (Deemed-to-be-University), Bangalore and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of Bachelor of Computer Applications in **CLOUD TECHNOLOGY AND INFORMATION SECURITY AND INFORMATION SECURITY AND MOBILE APPLICATION**. The results embodied in this thesis have not been submitted to any other University or Institute.

**Jasbir Singh [20BCAR0111] CTIS**

**Suman Garai [20BCAR0246] ISMA**

**Trishul V Biradar [20BCAR0146] CTIS**

**Umang Basantani [20BCAR0247] CTIS**

**Utsav [20BCAR0265] CTIS**

**ACKNOWLEDGEMENT**

It is not the completion of the project that is the most important but more so, the interaction of roles played by various people in the satisfactory completion of the same. We take this opportunity to express our deep gratitude and appreciation for all those who encouraged us to complete the project.

With a profound sense of gratitude and regards, we acknowledge with great pleasure the guidance and support extended by Dr. Suchitra Nair, Head, School of Computer Science & IT, Jain (Deemed-to-be) University, Bangalore for his interest and motivation towards the completion of this project.

We are grateful to our Guide, Dr. Taskeen Zaidi, Assoc. Professor, Department of BCA, Jain University, Bangalore, for her guidance and valuable time spent with us towards the successful completion of this project. We would also thank our parents, friends, and non-teaching staff for their support in completing this project.

**ABSTRACT**

As the number of online services grows, users need to control password management systems (generation, storage, retrieval). to have However, the absolute randomness and exclusivity requirements of passwords are not practical in everyday life. Each component of the password management system puts a cognitive load on the user. As users become more educated about the dangers of password reuse, the adoption of password managers at work and at home is also increasing. To help educate end users, we set out to prioritize security based on the most popular products. This study analyses popular password managers for features such as effectiveness, ease of use, and security that are considered essential to the security of the tool. These consist of features that affect user profile security for applications such as features that can compromise user security through multi-factor authentication and features which can affect user security through the security of the application such as the status of independent audits. We are also trying to find the ideal set of parameters to build the best possible password management system in 2022. This research helps us understand the key parameters and algorithms that can be used in designing the ideal password generation, storage, and retrieval system for users.

**TABLE OF CONTENTS**

**Acknowledgment (i)**

**Abstract (ii)**

**Table of Contents (iii)**

1. **Introduction……………………………………………………………..………………………..7**
2. **Problem Statement…………………………………………………………………….…………7**
3. **Project Objectives…………………………………………………………………………..…….8**
4. **Project Scope……………………………………………………………………………..……….8**
5. **Related Products……………………...………….……………………………………….….…...9**
   1. Literature Review
   2. Result & Analysis
6. **System Specifications……………………………………………………………………….…..17**
   1. Hardware configurations
   2. Software configurations
7. **Limitations of Existing Software’s……………………………...……………………………...18**
8. **Advantages of Proposed Software……………………………….………………..….……..….18**
9. **Economic Feasibility Assessment……………………………………………………..………..19**
10. **Requirement Analysis………………………………………………………………………..…19**
    1. Zero-Knowledge Encryption Architecture
    2. Database Synchronization
11. **Requirement Design & Implementations..……………………………………………….…....22**
12. **Test & Use Cases………....……………………………………………………………………...24**
    1. Database Access Security
    2. Encryption Security
    3. Password Generation Security
    4. Application Security Features
13. **Conclusions & Future Enhancements..………………………………………………………...26**
14. **References…………….……….…………………………………………………………………26**

**Introduction**

Security experts recommend password managers to help users generate, store, & enter strong unique passwords. Prior research confirms that managers do help users move towards these objectives, but it also identified usability and security issues that had the potential to leak data or prevent users from making full usage of their managers. A password manager, in the most basic sense, is a tool that saves a user's credentials (i.e., username and password) to reduce the cognitive load involved with remembering many unique login credentials. Password managers provide a mechanism for users to create, store, and fill strong, unique passwords. A password vault is a name given to this collection of passwords. The vault should ideally be stored encrypted, with the encryption key most typically generated from the master password, which is a user-chosen password. The password vault can be kept online if desired, allowing for synchronization across several devices. Because managers remember users’ passwords for them, users can generate strong passwords without the fear of forgetting or losing them. Most managers provide a generation pop-up dialogue next to password fields to encourage users to create strong passwords and periodically remind users if they still have weak or reused passwords. Moreover, research confirms that managers do reduce password reuse among their users. However, prior work identifies security challenges present in managers. The password autofill functionality may allow attackers to steal or phish users’ credentials. Additionally, password vaults occasionally leave users’ credentials or data vulnerable to attack, including unencrypted metadata and side channel leakage from encrypted information. No prior work examines the security of password generation in managers.

**Problem Statement**

In today’s world of technology, data holds the highest precedence. Data as in every action we commit in the Internet platform, be it posting photos in social media or messaging friends, remains as a meta-data of a user. Now, protecting these becomes a great responsibility from a privacy standpoint, since any 3rd party studying these can probably use those for their own benefits or even manipulating the user. Therefore, users need to carefully store the login credentials of any platform, so that no bad actors can get a hold of it and commit any cybercrime while pretending to be the user. And people these days

aren’t doing much to protect their login credentials. Entropy-based exclusive passwords from the point of security are a valid demand from many online services. But passwords that are difficult for an attacker to guess are also hard for users to remember, users often create weaker passwords to avoid the cognitive burden of recalling them. Despite the well-established problems facing password-based authentication, it continues to be the dominant form of authentication used on the web. Because passwords that are difficult for an attacker to guess are also hard for users to remember, users often create weaker passwords to avoid the cognitive burden of recalling them. Users, frequently employ easy tactics to get around these regulations, such as making minor changes to a popular password (e.g., adding a 1 to the end of a password used on another site). Users often make these adjustments using a limited set of simple guidelines, which can greatly increase an attacker's ability to guess passwords on other sites. In fact, with the increase in the number of passwords users are required to store, they often reuse passwords across websites. When it comes to password formation, human memory focuses on familiarity and repetition, putting us open to assault. People prefer to use the same password with personal information tacked on, according to research on building secure passwords done by Lo (2016). When requested to change a password, for example, adding a birthdate after the original password or adding the current month. Our minds can only carry around seven characters, so we can't store the long sequences of random characters required to be deemed safe in our internet world (Lo, 2016). Yan, Blackwell, Anderson, and Grant (2004) found that personal features such as a birthdate or pet's name are worthless against specific hacks in a survey of 288 college students. For example, if a hacker uses a dictionary attack to generate potential passwords using the same phrase, then variants of a password are just as simple to crack (Yan et al., 2004) The rejection of security advice by users is rational when the low percentage of users affected by breaches is contrasted with the effort required. However, the number of data breaches is on the rise, and this situation leaves many users vulnerable to exploitation. According to IBM’s Cost of a Data Breach 2021 report found that the average cost of a data breach for an organization was $4.24 million.

We thought something needed to be done about this. We propose a method for users to only need to remember one password that they use to access all their other passwords from any device at any time.

**Project Objectives**

The objective of this Password Management System is to manage passwords for different accounts on the internet. The user shall be able to save all the username and passwords information of the accounts he holds on using this application. These details shall be saved in the database in encrypted format. This will help the user to remember different usernames and passwords for accounts on the internet. The user shall be able to add databases, edit and delete databases using the system. The user has to login to the database, to use this tool. It takes user password, key file & hardware key. It allows the user to change the password for the database being used. This interface is developed using Android Studio and encryption systems like AES Rijndael, Two-Fish, ChaCha20. This Password Management System tool will allow the user to view all the different databases he / she holds in a list view on the screen. All the details related to the account like the user’s name, password etc. shall be displayed after opening the database. The password is visible in encrypted format. The user can unhide the password and view the password details. The user can save any account information using this tool, with the help of templates like Email, Wi-Fi, Notes, ID Card, Debit / Credit Card, Bank & Crypto Wallet. Since its offline focused, file-size won’t be a headache. However, you can store it online, to any of the popular cloud storage services.

**Project Scope**

* Saving Login Credentials into our Application.
* Using state-of the art cryptographic algorithms like Rijndael AES, Two-Fish, ChaCha20.
* Using proven key-derivation functions like AES & Argon2d.
* Every Login Credential saved in our application can be stored & sync in popular cloud services.
* Application will be secured with Master Password, along with Key File & Hardware Key options being available.
* Features like:
  + Auto-fill Services, Custom Private Keyboard.
  + Password / Passphrase Generator with Customization.
  + Screenshot Protection, Custom Favicon Support.
  + Write Protect, Auto-Timeout Lock, Auto Save Database changes,
  + Securely Storing Notes, Card Details, Identity Details, Email, Wi-Fi, Crypto Wallet.
  + Importing and Exporting Items in the Vault to be added.
* Options to Save New Credentials Prompt, Copy Password and TOTP for Filling, Clearing Clipboard.

**Related Products**

Literature Review

We did a comparative study of the existing Password Manager Solutions available in the market for the users based on a few criteria’s like, 2-Factor Authentication Support, Independent Audit, Open Source, Encryption Standard, IP Whitelisting, Self-Hosting Support Public Bug Bounty programmes, Password Generation Customization, Breach Alerts, Vulnerabilities Assessment etc.

Our investigation of many of the most popular password management offerings is useful for our education as it provides a comprehensive list of security features present so that we can select an offering which fits our needs. The results of this investigation have been categorised through checks and crosses on the following pages of readability where possible for the feature. Checks are representative of the features being present and the Crosses are representative of the feature being missing or unproven. We are also going to discuss in detail about the criteria’s we mentioned earlier and its implications in the applications.

1. *Encryption Standard*

In this criterion, we are going to analyse what encryption standards are being mostly used by the applications.

LastPass uses the master password that the user sets to generate the AES-256 keys which are then used to encrypt the user's 'vault'. The password is hashed 100,100 times with PBKDF2 SHA256 to generate the keys and then it is hashed an additional time before being stored as the authentication hash.

1Password uses AES-256 encryption for all data stored and the encryption key is stored locally on your device. Unlike LastPass, 1Password uses Two-Secret Key Derivation (2SKD) where the master password is only one of those secrets while the other is a cryptographically secure randomly generated string.

KeePass supports AES, TwoFish, and ChaCha20 for the encryption of any passwords stored in the tool, and with the chosen algorithm, it encrypts all data stored include usernames, passwords, or notes. SHA256 is used to hash the master password for authentication.

Dashlane utilizes AES-256 for encryption and decryption (Dashlane, 2020). Dashlane also does not store the master password in any form (Dashlane, 2019) which is used for the key generation with Argon2D and Dashlane exclusively deals with the AES encrypted data.

Bitwarden uses AES-256 encryption to encrypt all data related to the vault, and they use SHA256 to derive the AES keys from the master password. The encryption key related to decrypting the AES encrypted data is only stored for as long as the application is unlocked (Bitwarden, n.d.a).

Unlike some of the other password managers, NordPass uses XChaCha20 rather than AES for the encryption of the passwords (NordPass, n.d.a). It also uses zero knowledge architecture like other password managers where the encryption keys aren't stored in the infrastructure of the password manager developers. As well, NordPass doesn't store the master password and instead keeps the keys locally (NordPass, n.d.b). NordPass also allows the same encryption without an account locally (Walsh, 2020).

EnPass uses AES-256 with 100,000 rounds of PBKDF2-HMAC-SHA512. In laymen's terms, your passwords are encrypted with a unique "salt" value, repeatedly making your password harder to find with each of it's 100,000 iterations. EnPass encrypts everything locally on your device, "The key that encrypts your data is derived from your Master Password.

Keeper uses AES-256 to encrypt passwords and data at the device level prior to the encrypted data being sent to Keeper servers. Keeper also uses zero-knowledge architecture where the application doesn't store plaintext, Keeper infrastructure never receives plaintext, and the master password isn't stored by Keeper. As well, the AES keys are generated from the master password. For cloud authentication, two factor authentication and device verification must be completed prior to the AES keys being generated.

On their website, LogMeOnce claims to use AES-256 encryption to encrypt the passwords stored with LogMeOnce (LogMeOnce, n.d.a). No other information regarding their encryption could be found, and they only make their security whitepaper available in the administrator's package which can be considered a flaw compared to most other password managers.

Zoho Vault uses AES-256 for encryption of all private data (Zoho, n.d.a). The master password is not stored by Zoho and is used to generate the AES keys. Only AES encrypted data is sent over the internet and that's wrapped with TLS with strong ciphers for all connections (Zoho, n.d.b).

RoboForm uses AES-256 to encrypt all data with the keys remaining on the local device (RoboForm, n.d.d). As well, all data is placed into a single AES encrypted file instead of being stored separately (RoboForm, 2018). The AES keys are also stored by RoboForm, and while the private key is encrypted, it's encrypted using a key derived from the master password which is also stored along with the encrypted AES key as can be seen from figure 24 (RoboForm, 2018).

Keychain uses two different sets of AES-256 keys to encrypt the data, one being the table key, and the other being the per-row key. All data other than the kSecValue is encrypted with the table key to increase speed. The table key is protected by the 'Secure Enclave' but is cached to increase the speed further. The kSecValue is encrypted with the secret key and that key is not cached but instead must be fetched from the 'Secure Enclave' for each query.

All data stored and processed by Samsung Pass are encrypted using Samsung Knox's 'military grade' encryption standard (Samsung Knox, n.d.). Knox uses 'Dual Data-at-Rest' that simply encrypts your information twice using two separate keys that are stored in isolated areas of your device. The 'outer-layer' encryption uses the standard AES-256 while the 'inner-layer' encryption can be customized to include any third-party cryptographic module (or the default) to include 'security through individuality'. Knox's encryption framework has been certified by National Information Assurance Partnership (NIAP) as well.

1. *2FA / MFA Support*

Here we will talk about 2-factor authentication / multi-factor authentication support of different password managers available to users.

LastPass supports MFA in the free version of their platform (LastPass, n.d.c). Through that, LastPass supports the multifactor authentication options of LastPass authenticator, Google Authenticator, Microsoft Authenticator, Toopher, Duo Security, and Grid. These options can be found under Account Settings in the browser extension, and multifactor options under that. The options related to this can be found under Account Settings.

In 1Password, Two-factor authentication can be enabled on the device using Authy or Microsoft Authenticator. This can be configured under "Manage Two-Factor Authentication" under the user profile. After scanning the barcodes there with the preferred two-factor authentication application, it will be configured for the user's account.

Two-factor authentication can only be set up under the desktop application of Dashlane. The two-factor authentication for Dashlane can be configured to be required when you log into Dashlane from a new device, or every time you log into Dashlane.

Bitwarden supports two-factor authentication differently between premium and standard members. For a standard member, they can use an authenticator application or via email. For premium members, they can choose further between Duo Security, VubiKey, or FIDO U2F. For a standard member, the two-factor authentication can be configured under Account.

NordPass supports a variety of two factor auth apps through scanning a QR code, and the examples it gives is Google Authenticator, Duo, and Authy. This can be set up after connecting the app to your Nord account, going to settings, and two-factor authentication and scanning the QR code.

Two-factor authentication can be configured for Keeper, through the Keeper vault and offers methods such as text message, Microsoft or Google Authenticator, Duo Security, Yubikey, among others.

LogMeOnce supports a variety of two-factor authentication options which can be configured under the general settings. Some of these options are Selfie 2FA, SMS Message, Voice Call, Email message, TOTP, USB Flash Drive, X.509 Certificate & Security Key.

Zoho Vault offers multi-factor authentication through text message, voice call, Yubikey, Google Authenticator or Zoho OneAuth.

RoboForm supports two-factor authentication through text message, email, or Google Authenticator. This can be configured under security settings after the user signs into their account.

Two-factor authentication can be configured for the user's Apple ID (Apple, n.d.a) and Keychain requires two-factor authentication to use the same device passcode to recover a Keychain (Apple, 2021b).

As Samsung Pass is an application that is pre-installed onto mobile smart phones, Samsung has several options when using MFA including biometric options as well as a simple text message verification.

1. *Mobile Application Biometric Support*

For developing a mobile application, its necessary to ensure best possible security with device hardware support wherever possible.

LastPass supports biometrics in the mobile application (LogMeln, n.d.) and configuring biometrics is prompted right after installing the application. After the first login, biometrics can be turned on under security under settings, and prompts you to scan your fingerprint after turning it on. LastPass also offers account recovery with biometrics.

1Password uses biometric authentication in the Android version of their mobile application. Specifically, they utilize Android's biometric authentication, so no biometric information is stored by 1Password (1Password, 2020a).

While KeePass doesn't officially support the Android port, the Android port of KeePass supports biometric authentication (Knight, 2020).

Dashlane supports biometric authentication and enables it automatically after installation. If users miss or disable it, it's still available to be configured under the security settings.

Biometrics are supported in both Android and iOS applications for Bitwarden and can be configured in the settings.

NordPass supports fingerprint authentication in the mobile application (NordPass, n.d.c) and it can be configured in the settings menu.

EnPass supports biometrics in the mobile application, and it can be configured under security settings through the quick unlock section (EnPass, n.d.b).

Keeper supports touch and face ID for iOS, and fingerprint for android for authentication.

Those can be configured through the settings (Keeper, 2021b).

LogMeOnce supports biometric authentication through fingerprint support and can be configured through the settings.

Zoho Vault supports biometric authentication for the mobile application through the "Swift Login" setting which will bypass the master password authentication.

RoboForm allows biometrics to be used for authentication for the mobile application and this can be configured through the "Authorized Access Options" (RoboForm, n.d.a). Unrelated to the mobile application, RoboForm also supports authentication in the desktop application through Windows Hello if Hello is configured through Windows (RoboForm, n.d.b).

iCloud Keychain can be configured to use touch or face ID (Apple, n.d.b).

Samsung Pass allows biometrics being used to authenticated in order to fill in passwords and IDs (Samsung, n.d.)

1. *Password Generation*

Password managers a now-a-days has become a lot more than only credentials storage and helps its users to create a strong password for its accounts.

LastPass allows a length of 0-99 with options such as easy to say which only uses letters, easy to read which can include letters, numbers and symbols, and will avoid confusing combinations such as 1 and I. All characters use the selection of letters, numbers, and symbols, and doesn't follow any other rules.

1Password offers their password generator online, even without an account for 1Password. The passwords generated are exclusively random with no option to add existing words. The password can be up to a length of a hundred characters composed of letters, numbers, and symbols. There are also the options for memorable password and pin on top of the random password. Pin offers the option of a three to twelve number string, and memorable password offers the option of three to fifteen words separated by dashes.

KeePass offers a password generator with the length controlled by a text box which doesn't appear to have a hard limit. The password generator includes letters, numbers, and symbols. It also offers the ability to create passwords based on patterns which offers an extensive number of options (KeePass, n.d.c).

Dashlane's password generator offers a length from four to forty with numbers, letters, and symbols.

Bitwarden offers a password generator that can generate a password with the length of five to a hundred and twenty-eight characters. Users can freely choose the password style they want, including whether this is a password or a passphrase. Users can also select its characteristics, including capitalization, and whether it contains numbers and symbols as well as the number of times that they can appear in the password.

NordPass offers a password generator which creates a password with a length of between eight and sixty characters. It supports letters, numbers, and symbols as well as the option to avoid ambiguous characters.

EnPass' password generator allows the user to generate their password with many different criteria, including the specific special characters allowed as well as the number of special characters, numbers, and uppercase letters used. The tool can create a variable length password with a minimum of four characters and it offers a variety of options such as generating the password using real words.

Keeper offers the random password generator which generates a random password with a long length of letters, numbers, and symbols.

LogMeOnce offers a password generator which can create a password with a length of six to fifty characters comprised of letters, numbers, and symbols. A nice feature of LogMeOnce's password generator is adding the estimated time to crack the password after hashing which might be more intuitive to a regular user than the bits or more practical than a classification such as 'secure'.

Zoho Vault offers a password generator that creates passwords with a length of four to a hundred characters. Zoho also offers the options of numbers, special characters, and letters. Additional options include starting with a letter and mixing the case of letters. While the password generator is comprehensive, the user cannot customize the number of occurrences of special characters, and it may take several times to get a relatively secure password.

RoboForm offers a password generator which can generate a password with a variable length dictated by a textbox, much like KeePass. The default length of a generated password is sixteen characters with letters, numbers, and symbols. RoboForm has additional options such as utilizing hexadecimal in the password generation as well as deciding which special characters are included.

iCloud Keychain offers a password generator with a variable length and multiple types (iClarified, 2012).

1. *Breach Alerts*

Getting to know from a Password Manager itself, if its users’ credentials are available in the Darknet and suggestion to change the credentials of the same account is common in today’s password managers.

LastPass does not support breach alerts in the free version, but it does support breach alerts on the premium and above versions. The feature is coined as "Dark Web Monitoring" (LastPass, n.d.e) in those versions and it alerts users by email whenever any emails that they use are detected in password dumps provided by Enzoic (Bachmann, 2020).

1Password comes with the feature of "Watchtower" (1Password, n.d.) which utilizes the HavelBeenPwned API to check if any of the passwords used by the user have been detected in passwords dumps. The notifications tab for watchtower is directly on the homepage when you log in to the application, and it will list off all the compromised websites as well as give you options for changing your password.

Dashlane provides a feature called "Dark Web Monitoring" on premium versions of the application and monitors emails that are submitted to it. The feature takes a maximum of five email addresses and checks password dumps which contain those emails as part of the credentials (Dashlane, n.d.b).

Bitwarden has a feature for password checks called the "Data Breach Report" which utilizes HavelBeenPwned as well to check any password dumps which contain the used password (Spearrin, 2017a). Currently, the feature is available on the free version when used with one password at a time.

NordPass has the feature of "Data Breach Scanner" in the premium version of NordPass but does not provide an equivalent feature for the free version. NordPass' "Data Breach Scanner" checks dumps against any stored data including email addresses, passwords, and credit card numbers (NordPass, n.d.d).

EnPass uses HavelBeenPwned as well and this can be checked by the user through the audit tab on the front page of the application. The application checks the stored passwords with password dump contents locally and alerts the user if any passwords have been compromised.

Keeper does not support breach alerts natively in the free version, but they do support them through the paid business addon of "BreachWatch" (Keeper, n.d.b). Keeper automatically alerts the user whenever a password that is used is detected in a breach, and they claim that they track over a billion known passwords.

LogMeOnce provide breach alert features such as leaked password monitoring, dark web monitoring, and anti-theft but those features require the user to purchase them. Dark web monitoring does provide one free scan to new free users, however.

1. *Independent Audit*

Independent Audits unilaterally increases trustworthiness of any password manager being used by many.

1Password has been the focus of multiple audits, assessments, and pen-tests, all of which are publicly available on their website (1Password, 2021a). Their most recent Audit was from Onica who frequently provide security audits to businesses that incorporate AWS, such as 1Password. Onica stated that there were no high-risk issues found during the security audit. Their most recent pen-test from Cure53 found 2 medium risk threats stating that the 1Password Vault that stores the user's passwords are vulnerable to compromise and these issues have yet to be remediated.

KeePass was audited by the European Commission's Free and Open-Source Software Auditing project (KeePass, n.d.b) in 2016, and published the full results (Everis, 2016). The full results shows that there were no critical findings but there were five medium findings.

Dashlane claims across their website that their products are audited regularly and by different security firms (Dashlane, n.d.a) but we were unable to find which companies audited the products or any information on the audit. Other reviewers were unable to find public information regarding independent audits as well (Long, 2020) except for an investigation conducted by MIT students in 2016 which found no vulnerabilities (Gentili et al, 2016).

Bitwarden undergoes independent audits including an audit by Cure53 (Dashlane, n.d.b) which has it's results publicly posted by Bitwarden in a security assessment report (8Bit Solutions LLC, 2018). Bitwarden also updates this security assessment regularly with the last update being in 2020 with two reported issues (Bitwarden, 2020).

NordPass underwent an independent audit by a Cure53 team lead by Dr. Mario Heiderich and published a summary of the results on their NordPass domain available to those that have Nord accounts (NordPass, 2020). Cure53 is an incredibly reputable auditing firm that holds privacy highly, even going so far as to have auditors contribute to University of Toronto's citizen lab on a project investigating the privacy of South Korean monitoring apps (Kenyon, 2017).

VerSprite conducted a security audit on Enpass' applications (Leo and Watson, 2018), published it on November 30th, 2018, and concluded that the overall risk impact was "medium", as 2 vulnerabilities (1 in their Windows desktop app, and 1 in the Android app) were found that exposed the users unencrypted master password. The Windows vulnerability was remedied before publication of the assessment; however, the android version was only partially remediated as an attacker may still gain access to the master password given root privileges. Source code and credentials were provided to VerSprite for the purpose of this assessment. They are currently looking into another security assessment, but no details have been provided at this time.

Keeper claims to be audited regularly and claimed in an email response to our security inquiry that an NDA is required for sharing the auditing results. The companies they claim independently audit their product include the NCC Group, Secarma, Rhino Security, and Cybertest.

1. *Public Bug Bounty*

LastPass has a public bug bounty program (https://bugcrowd.com/lastpass) which includes the LastPass product, the browser extensions, the desktop applications, and the mobile applications in the scope of the program. Some important out of scope targets/vulnerabilities include 2FA/MFA mobile applications, credentials being leaked through memory dumps, desktop applications being exploited by a malicious application or browser extension, and man-in-the­ middle attacks.

1Password has a public bug bounty (https://bugcrowd.com/agilebits) which focuses on the website, signup process, authentication, features included in the web application, as well as their native apps. However, they do not give rewards for bugs that abuse scheduled infrastructure changes, headers used for maintaining session ID's, DoS/DDos attacks, or any attacks that require root access. This leaves several attacks unchecked such as Session Fixation, Session Hijacking, and Cross Site Request Forgery.

It was said that in January 2019, the European Union was to fund KeePass for the bug bounty program (Cimpanu, 2018). No other recent reports regarding if this went forward could be found.

Dashlane has a public bug bounty program (https://hackerone.com/dashlane?type=team) hosted on Hackerone which covers vulnerabilities in features/products such as autofill/autologin, their website, API endpoints, client applications on different platforms, and standalone extensions. On the other hand, Dashlane uses only in scope products for their bug bounty with a mass exclusion on anything not specifically stated as in scope which would then exclude any vulnerabilities that rely on Dashlane being unaware of their existence such as subdomain takeovers or legacy endpoints found through Shodan.

Bitwarden has a bug bounty program hosted on Hackerone(https://hackerone.com/bitwarden?type=team) which unlike Dashlane, has a mass inclusion where they include any vulnerability affecting Bitwarden products in the scope with the exclusion of certain vulnerabilities. These vulnerabilities include issues such as self-XSS, username/email enumeration, or attacks that require physical access to a user's device. They also handle, without a bounty, any issues submitted to tickets on Github (https://github.com/bitwarden).

NordVPN, the parent company of NordPass, has a bug bounty program but they explicitly exclude any infrastructure related to NordPass from the program scope which we can consider to be harmful to the bug bounty status of NordPass.

Keeper Security has enacted a Bug Bounty Program. Keeper's mission is to build the world's most secure and innovative security apps, and therefore they believe that bug reports from the worldwide community of security researchers is a valuable component to ensuring the security of Keeper's products and services. In order to submit a report, Keeper requires it to be done through Bugcrowd. Some of the guidelines under their policy include: to be authorized in accordance with Computer Fraud and Abuse Act, consider it exempt from DMCA, and will not bring a claim against you for bypassing any security or technology controls, consider it legal, and will not pursue or support any legal action related to this program against you, will work with you to understand and resolve the issue quickly, will recognize your contributions publicly if you are the first to report the issue and will make a code or configuration change based on the issue, avoid violating privacy, harming user experience, disrupting production or corporate systems, and/or destroying data, perform research only within the scope, contact Keeper immediately if user data is encountered during testing, use identified communication channels to report vulnerability, and lastly keep information about any vulnerabilities confidential until Keeper has resolved it.

LogMeOnce has a public bug bounty program which is hosted under their vulnerability disclosure policy (LogMeOnce, 2020). Their reward amounts are far below other password managers bug bounty programs with thirty dollars for a low severity vulnerability and fifty dollars for a medium severity vulnerability. However, they do acknowledge that reward amounts may change. They also have a long list of out-of-scope vulnerabilities, and while a lot of them are reasonable to be out of scope, there are some serious scope exclusions such as reflected file download, link injection in LogMeOnce sent emails, and being able to bypass file extension whitelist.

Zoho has a complete self-hosted bug bounty system of its own (https://bugbounty.zoho.com/bb/info?department=zoho#scope). In scope is any Zoho branded products, applications listed at their main site, all Zoho branded mobile applications, and all Zoho branded client-side applications. This scope would then include Zoho Vault. Zoho also has a fairly long list of exclusions, but those are just exclusions with regard to the bounty payment as they state they may acknowledge you in the hall of fame in leu of payment for excluded vulnerabilities.

Apple has enacted a security bounty (https://developer.apple.com/security-bounty/). Apple's commitment to security includes rewarding researchers who share with them critical issues and the techniques to exploit them. Apple offers public recognition for those who submit valid reports and will even match donations of the bounty payment to qualifying charities. In order to be eligible for an Apple Security Bounty, the issue must occur on the latest publicly available versions with a standard configuration and, where relevant, on the latest publicly available hardware or the Security Research Device. Other than that, the Apple bug bounty is extremely inclusive with the vulnerabilities it accepts.

Samsung has bug bounty program for Samsung mobile devices rather than Knox specifically, but the only software excluded from the scope is third-party software, so Knox qualifies under this program (https://security.samsungmobile.com/rewardsProgram.smsb). Some important excluded vulnerabilities include vulnerabilities that require physical connection to the device with a debugger, scenarios which require a high amount of user interaction, or any exploit where the probability of exploitation is very low. However, they don't define 'very low' within the program so this might discourage critical vulnerabilities from disclosure if it requires extensive exploitation.

1. *Past / Present Security Vulnerabilities*

Now, we look at the records of the Password Managers available in the market, and whether there has been or is any fiasco regarding storing of users’ credentials.

LastPass has six previous CVEs going back as far as 2013 with two 2020 CVEs which are disputed due to the vulnerability relying on a jailbroken device (Mitre, n.d.a). There is also a public denial of service exploit from 2015 which is a basic stack-based overflow exploit in the login screen (Exploit Database, 2015).

There is a 2021 CVE for 1Password for a vulnerability which allowed anyone who had the authentication for a business's SCIM Bridge platform to view the TLS private key for securing connections to the bridge over the internet (1Password, 2021c). There is also a 2020 CVE for 1Password which revealed that 1Password's SCIM Bridge platform was using insecure RNG to generate keys. This allowed potential brute force attacks (1Password, 2020b).

There are seven CVEs for KeePass going back as far as 2010 with two related 2020 CVEs which lead to attackers being able to read or modify data in the KeePass database (Mitre, n.d.b). There are also two public denial of service exploits for KeePass (Exploit Database, 2016) including an exploit from January 2020.

There is one previous CVE for Dashlane from 2017 related to DLL hijacking (Mitre, n.d.c) and there is also a public exploit for the CVE (Exploit Database, 2017).

There are two previous Bitwarden CVEs including a 2020 CVE related to server-side request forgery and a 2019 CVE related to a potentially unwanted KDF (Mitre, n.d.d). There is also a possible remote code execution vulnerability through Bitwarden's auto-update (Github, 2020).

There aren't any present or previous vulnerabilities related to NordPass but there are previous CVEs and public exploits for NordVPN. The CVEs for NordVPN include previous vulnerabilities for code execution and local privilege escalation (Mitre, n.d.d). The public exploits available for NordVPN include a public exploit for an unquoted service path from 2020 which may also occur in NordPass as unquoted service paths are common (Exploit Database, 2020).

There are two important previous vulnerabilities for EnPass including a 2017 vulnerability which could cause EnPass to be the target of a local file inclusion attack (EnPass Team, 2017) and lead to code execution. There was also a 2020 vulnerability which could lead to the injection of malicious code (EnPass, n.d.c).

Some vulnerabilities were found related to RoboForm by researchers Michael Carr and Siamak Shahandashti including PIN bruteforcing and clipboard data theft (Wagenseil, 2020). According to Fortinet researchers, there has also been a zero-day vulnerability for RoboForm pending a fix since 2018 (FortiGuard Labs, 2018) which could be a serious security concern related to the product depending on the zero-day vulnerability.

There are many CVEs related to Keychain or obtaining Keychain items going back as far as 2004 with the latest vulnerability being reported in 2018 (Mitre, n.d.e).

A vulnerability for Samsung pass was found in 2019 and reported to the Zero-Day Initiative that allowed attackers with physical access to a device to gain access to sensitive data within the Samsung Knox secure folder, or the "vault" that contains all your Samsung Pass data (Zero-Day Initiative, 2019).

1. *Miscellaneous Notes*

Other than the criteria’s discussed already, we are going to look at a few more aspects regarding the Password Managers we have been discussing for so long.

LastPass does not necessarily support IP whitelisting for the password manager natively, but they do support IP whitelisting through the paid 'Identity' service (LastPass, n.d.d). At the time of writing this report, it has been discovered by Mike Kuketz that LastPass has seven tracking features on the Android application (Owaida, 2021). There's no reason why a password management application would need to track users, and this is a large privacy concern.

1Password supports IP whitelisting through 1Password for business but they do not support the feature in the regular version of the password manager (1Password, 2021b).

KeePass does run under an open-source software. It is a free open-source password manager. Passwords can be stored in an encrypted database, which can be unlocked with one master key. The open source is OSI certified. Database files are encrypted using the best and most secure encryption algorithms, such as AES-256, ChaCha20, and TwoFish (KeePass, n.d.a). KeePass's storage method is entirely self-hosted. The KeePass password vault is stored locally without support for any cloud hosting, but that can be setup to be cloud hosted by the user through a platform such as Dropbox.

Dashlane is not open source, but they do have an active Github where some code can be found such as the iOS Extension (https://github.com/Dashlane).

Bitwarden is an open-source password manager and hosts all code related to functions such as the server, the web version, the desktop application, and the mobile application on Github (https://bitwarden.com/open-source/). Bitwarden's primary method of managing passwords is through local storage without cloud support. Bitwarden also offers the ability to host the password manager through Docker (Spearrin, 2017b) which would lead to the password manager being self-hosted.

NordPass allows the user to host it themselves through a local password database, and it doesn't necessarily require a Nord account to use it.

Keeper does not support IP whitelisting in the free version and doesn't natively in other versions. However, it is possible the IP whitelisting could be configured through active directory in the business version (Keeper, n.d.a).

LogMeOnce offers the ability to freeze your account including freezing access from any IPs other than the users (Support Team, 2018) but this couldn't be replicated by our team and appears to be a part of LogMeOnce which requires a monthly fee (LogMeOnce, n.d.b).LogMeOnce allows the user to change the storage mode that they are using. They support the storage modes of local only and cloud storage. When storing locally only, the password manager would be self-hosted by the user.

Zoho Vault supports IP whitelisting in the standard, professional, and enterprise versions which require being purchased but not the default free version of Zoho Vault (Zoho, n.d.c).

RoboForm does not have IP whitelisting in the free version. However, they do offer the functionality in their business version (RoboForm, n.d.c). RoboForm allows the ability to use local only storage for the password file in both the desktop and mobile applications (RoboForm, n.d.d).

Keychain is open-source software released under the terms of APSL and can be found on Apple's open-source\_subdomain(https://opensource.apple.com/source/libsecurity\_keychain/Libsecurity\_keychain-27723/).

Result & Analysis

Existing password managers that employ genuinely random password strings offer strong authentication. Most contemporary systems provide users the choice of maintaining their existing non-random credentials or setting new random passwords during registration.

KeePassXC provided the most complete symbol set, containing both the extended ASCII symbol set and all the normal ASCII symbols (space excluded). Apart from space, only the conventional ASCII symbol set is supported by KeePassX and Dashlane, not the expanded ASCII symbol set. Only 19 ASCII symbols are supported by 1Password, compared to the other systems' acceptance of no more than 8 of them. The strength of passwords drastically decreases when the symbol set is limited.

The second phase of the password manager lifecycle is password storage. Both extension-based and app-based password managers use AES-256 encryption for their databases. App-based and extension-based password managers now have a substantially greater ability to safeguard metadata. All metadata is encrypted by both KeePassXC & all other Password managers we surveyed.

Password managers employ a variety of techniques to store and automatically fill in passwords. For instance, cloud data can be copied locally and encrypted with a master password using app-based password managers.

If a password manager autofill’s password without asking the user first, the user's password can be obtained by visiting a website that has been compromised. Therefore, before autofill happens, user interaction should be necessary. Although this functionality is deactivatable, KeePassXC, Bitwarden, and RoboForm all require user interaction by default.

Regardless of user involvement, passwords that are automatically put in within iframes are extremely risky. For instance, clickjacking allows an attacker to acquire passwords for vulnerable websites loaded in an iframe by tricking visitors into providing the information necessary to automatically fill in their login credentials. Even worse, if user involvement is not necessary and cross-domain iframe autofill is permitted, the attacker may automatically collect the user's login information for any website where a network injection or XSS attack may be conducted.

There are numerous methods for making, keeping, and remembering passwords. It is nevertheless suggested to choose passwords that have a suitable length because even randomly selected weak passwords are likely to be immune to online and offline attacks. According to our analysis, there is a 10/10 ratio between online and offline assault resilience. The accepted method for storing local databases and information is AES-256 & PBKDF2 for transporting it to the cloud, although other methods (mainly used by KeePassXC) are similarly secure. All OS-based mobile autofill frameworks demand user interaction prior to autofill, and they all guarantee security and accessibility for data recall. Furthermore, iOS password autofill completely encrypts the autofill process for native UI components in apps. Local storage with master password encryption is thought to be the best approach for web extensions.

**System Specifications**

Hardware configuration

*Deployment Platform*

* Chipset: Apple A6 (32nm) & Qualcomm MSM8974 800 (28nm)
* CPU: Quad-Core 2.3Ghz Krait 400 & Dual-Core 1.3 Ghz Swift (ARM v7 based)
* GPU: Adreno 330 & PowerVR SGX 543MP3 (Triple-Core Graphics)
* Memory: 1 GB eMMC 4.5

*Development Platform*

* CPU: 1.8 GHz or faster 64-bit processor; Quad-core or better recommended.
* Memory: 4GB (min) 16GB (recommended)
* Hard Disk: 20 to 50 GB
* GPU: Nvidia GTX 960 and above

Software configuration

*Deployment Platform*

* Operating System: Android 8.0+
* Size on Disk: Approximately 50 MB

*Development Platform*

* Android Studio Dolphin // 2021.3.1
* Kotlin / Java / JNI / C
* Assets from Google

**Limitations of Existing System**

In most cases, Password Management is done manually. Users must remember the list of passwords for different accounts on the internet. The user must manually maintain a list of all usernames and passwords. This task was very tedious. In cases where privacy for user accounts was required, it was difficult to manually maintain this list of passwords. The user had to save password details list in some word format so that it would help him to remember the username and passwords for different accounts on the internet.

Other than that, today’s market is flooded with Password Managers, mostly paid and some free. From reviews from various blogs and personal experience, we learnt that the most popular ones had a few major drawbacks.

* LastPass, its plagued with autofill issues, outdated apps, limited free version features and 2022 data breach adds a black spot to its otherwise feature rich services.
* Dashlane, another service provider, also has limits to its password storage options, simultaneous devices usage features and cloud storage.
* LogmeOnce, a great freemium, is overloaded with features and user-interface and experience is clumsy, thus making it overwhelming for common users.
* BitWarden, an open-source solution, lacks TOTP, Password Sharing features with its free versions. KeepassXC, another totally open-source option, lacks password sharing. Moreover, configuring them manually can be a headache.
* 1Password, NordPass and Keeper, probably the leaders in this space of password managers, lacks free versions, which is a setback to people unwilling to spend a buck on these.
* RoboForm, a popular name being relatively inexpensive, lacks TOTP support for its mobile apps along with no cloud sync options.

**Advantages of Proposed System**

That’s where our software comes into action. We plan to make the most comprehensive Password Management System Graphical User Interface, developed using Android Studio. It allows the user to add credentials for different accounts he holds on to the internet. Password Management System not only stores the user’s name and password details in an encrypted format in the database, but also a lot more types of sensitive data. The encryption is managed by Rijndael AES, Two-Fish & ChaCha20 algorithms. The user has to login to the system. It checks for the validity of the user. The user shall be able to add / create a database. The user can add username and password details, according to the Templates provided and save to the database. The list of accounts held by the user is displayed under groups, if organized accordingly. When the user selects the account on the screen, the details of the account get displayed opens. The password shall be displayed in hidden format. Once the user selects the un-hide option, he shall be able to view the password on screen. The user can add any number of accounts using this application. The Key-Derivation Function is very robust and uses the Argon2 / AES-KDF, which is a very effective way of handling passwords. We also plan to add features like:

* Zero-knowledge Encryption,
* Unlimited Vault Storage & Sync,
* Open-Source Codebase,
* Secure Password & Passphrase Generator,
* 2-Factor Authentication Login,
* TOTP & HOTP Support for Stored Credentials,
* Store notes, credit cards, identities, and many more.

while keeping the cost as low possible so that more people can take the benefits of using a proper management for all their online credentials, and not compromise on their security.

**Economic Feasibility Assessment**

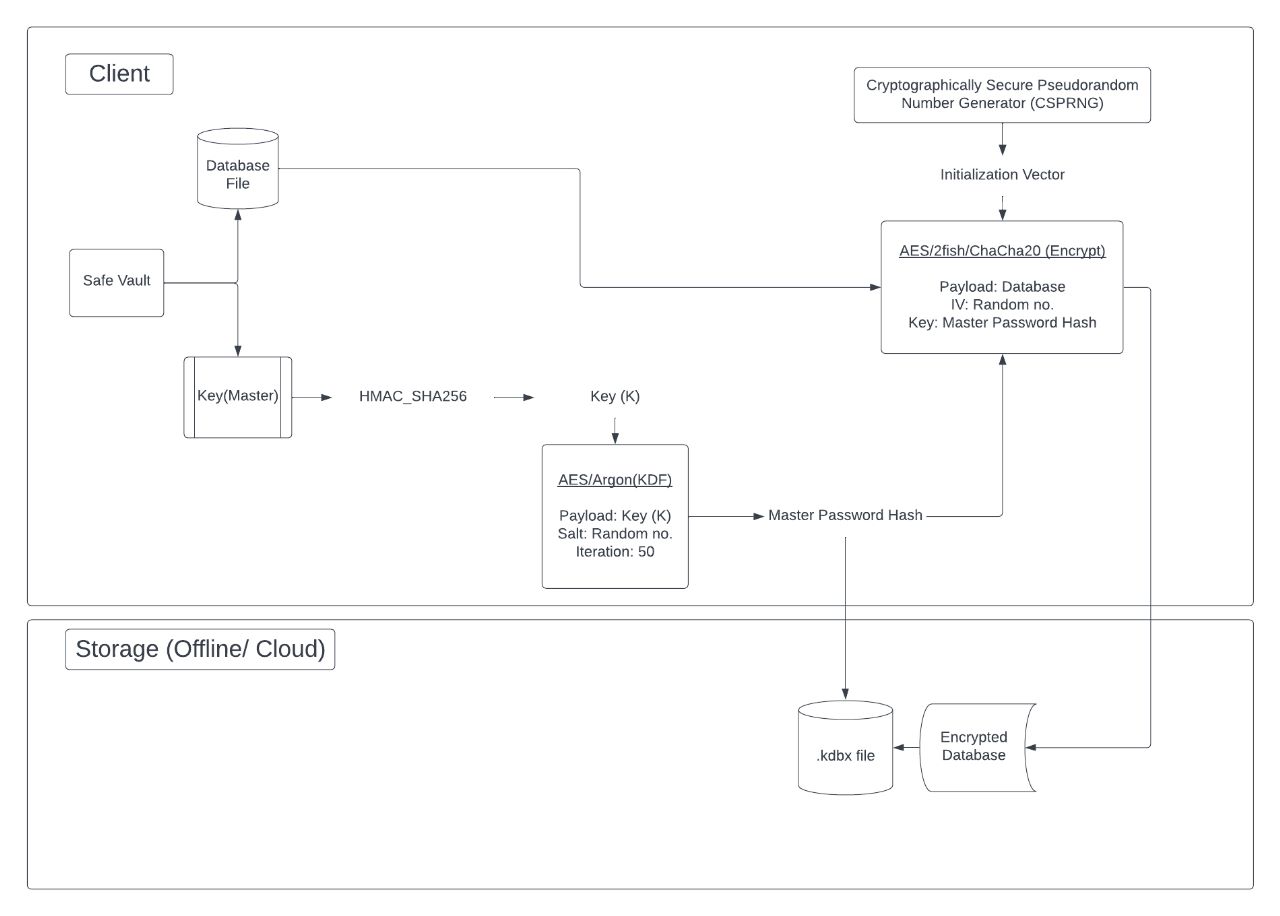
The proposed system will mainly be developed on software’s such as Android Studio, which is freely distributed online by Google Inc., and programming languages like Kotlin & Java will be used, in accordance with what we learnt from in our higher studies till date. Hence no cost will be incurred for those.

The deployment platform consists of Android Devices which is widely available with all the members. We also have laptops that meet the requirements, as the development platform. For cloud server storage, we leave it up to the users, as our solution will be compatible with a lot of solutions like Google Drive, OneDrive, Dropbox etc. till we have the said cloud storage providers app installed in our smartphones along with our app.

Overall, it’s just the internet cost that we are paying for downloading the required tools & for cloud storage synchronization. Other than that, our project is cost free, until we consider our labour cost.

**Requirements Analysis**

Zero-Knowledge Encryption Architecture



*Database Encryption*

The database files are encrypted. KeePass encrypts the whole database, i.e., not only your passwords, but also your usernames, URLs, notes, etc.

The following encryption algorithms are supported:

These well-known and thoroughly analyzed algorithms are very secure. AES (Rijndael) became effective as a U.S. federal government standard and is approved by the National Security Agency (NSA) for top secret information. Two-Fish was one of the other four AES finalists. ChaCha20 is the successor of the Salsa20 algorithm (which is included in the eSTREAM portfolio).

The block ciphers are used in the Cipher Block Chaining (CBC) block cipher mode. In CBC mode, plaintext patterns are concealed.

An initialization vector (IV) is generated randomly each time a database is saved. Thus, multiple databases encrypted with the same master key (e.g., backups) are no problem.

The authenticity and integrity of the data is ensured using a HMAC-SHA-256 hash of the ciphertext (Encrypt-then-MAC scheme).

*Key Hashing and Key Derivation*

SHA-256 is used for compressing the components of the [master key](https://keepass.info/help/base/keys.html) (consisting of a master password, a key file, a Windows user account key and/or a key provided by a plugin) to a 256-bit key *K*.

SHA-256 is a cryptographic hash function that is very secure. It has been standardized in [NIST FIPS 180-4](https://dx.doi.org/10.6028/NIST.FIPS.180-4). The [attack against SHA-1](https://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html) discovered in 2005 does not affect the security of SHA-256.

To generate the key for the encryption algorithm, *K* is transformed using a key derivation function (with a random salt). This prevents precomputation of keys and makes dictionary and guessing attacks harder.

*Protection against Dictionary Attacks*

Such attacks cannot be prevented, but they can be made harder. For this, the key K derived from the user's master key is transformed using a key derivation function with a random salt. This prevents a precomputation of keys and adds a work factor that the user can make as large as desired to increase the computational effort of a dictionary or guessing attack.

Multiple key derivation functions are supported.

The key transformation may require more or less time on other devices. Make sure that all devices are fast enough (and have sufficient memory) to load the database with your parameters within an acceptable time.

Supported key derivation functions:

AES-KDF:  
This key derivation function is based on iterating AES.

In the database settings dialog, you can change the number of iterations. The more iterations, the harder are dictionary and guessing attacks, but also database loading/saving takes more time (linearly).

Argon2:   
[Argon2](https://github.com/p-h-c/phc-winner-argon2#argon2) is the winner of the [Password Hashing Competition](https://www.password-hashing.net/). The main advantage of Argon2 over AES-KDF is that it provides a better resistance against GPU/ASIC attacks (due to being a memory-hard function).

The official specification of the Argon2 algorithm defines three variants: Argon2d, Argon2id and Argon2i.

Argon2d provides the best resistance against GPU/ASIC attacks. The resistance of Argon2id against GPU/ASIC attacks is somewhat weaker, but Argon2id additionally makes certain side-channel attacks slightly harder. The IRTF CFRG Argon2 Internet standard recommends Argon2id by default. For server applications, Argon2id is in general indeed more suitable than Argon2d.

Note: We currently recommend Argon2d instead of Argon2id, because we believe that a better protection against an existing threat (password cracking using GPUs/ASICs is state of the art) is more important than a protection against certain side-channel attacks that may or may not become a problem on client devices in the future. If you worry about side-channel attacks (and are willing to sacrifice some GPU/ASIC resistance) or if you are developing a software where side-channel attacks could be a problem (e.g., a server service that operates with KeePass database files), use Argon2id.

Database Synchronization

SafeVault does not have an internal database file copy to synchronize data. This allows for greater scalability and configuration; no data is stored in the application when the database is closed.

With this system, a database file can be :

1. stored and used locally to the device.
2. stored on a medium external storage to the device (ie: [OTG](https://en.wikipedia.org/wiki/USB_On-The-Go)). If the device is lost, no data, even encrypted, can be recovered.
3. provided by any remote file manager (cloud app).

The file can therefore be local or remote depending on your needs and the file manager used.

Synchronization can therefore be seen as a two-step action:

1. Data recovery from a predefined shared location. It is not up to SafeVault to handle this step, because the architecture of the Android system is built in such a way that a dedicated application manages to send and receive the files and then propose them through a [URI](https://en.wikipedia.org/wiki/Uniform_Resource_Identifier). Here, our app is only an editor and file synchronization problems might be caused by the cloud application used because it doesn't handle file conflicts or has caching problems.
2. The merge of the updated data. When the URI provides new data, SafeVault can merge the currently open data with this retrieved data.

Diagram

Description automatically generated

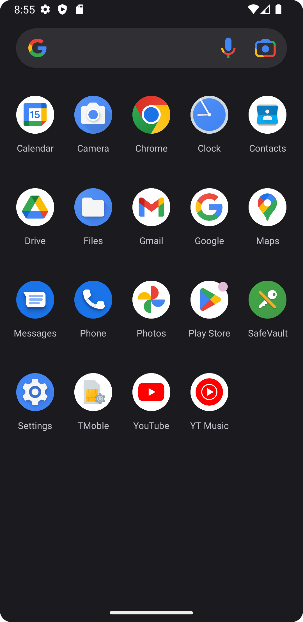
There are 3 commands in SafeVault to manage the synchronization:

* Save data: As the name suggests, it allows you to save the local data of the open database in the open data stream from its URI. This is the conventional method used to update the data on the reference medium.
* Merge data: Retrieves the data from the URI and merges it with the local data. This command does not ask for a credential, if you change the main password, use Merge from.... Please note that this command does not automatically save the database even if the automatic database save setting is enabled. You will have to save the database manually after a merge if that is what you want.
* Reload data: Retrieves data from the URI and deletes local data. Use this method if you want to reset the data in your database with the data in the storage medium.
* Merge from... : Performs a merge with the same characteristics as the Merge data command but with a different URI file. The main credentials of the database to be merged will always be requested. It is possible to merge databases of different versions with this command. (For example: possible to merge a .kdb v1 database into a .kdbx v4 database)
* Save a copy to... : Saves current data to another specified file (URI). The data of the current URI are not modified.

**Requirement Design & Implementation**

In this section, we are dealing with how to work with our application in Android Devices.

Firstly, we will install the app & see something like this icon in our Home screen of Android Devices.



A screenshot of a computer

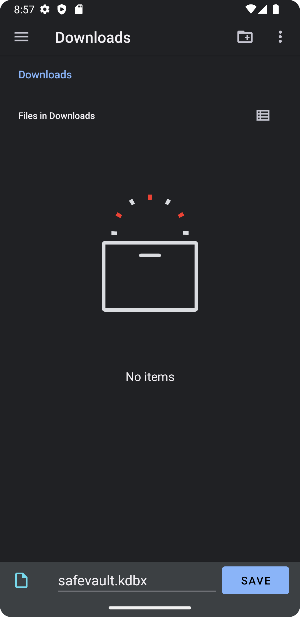
Description automatically generated with medium confidenceIn this Screenshot, from the right-hand side 1st column 3rd row, we see our installed app icon. Clicking and opening our app will give us something like this in our screen.

Here, after the splash screen, we see the opening screen

of our application. There is already a database created

with its details provided accordingly. We will start with

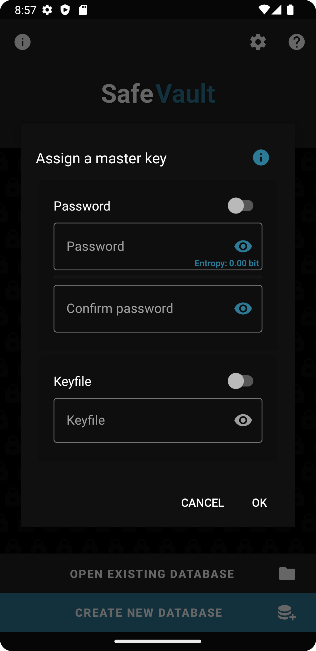
creating another one.



The Files Manager of the Android system opens. Here

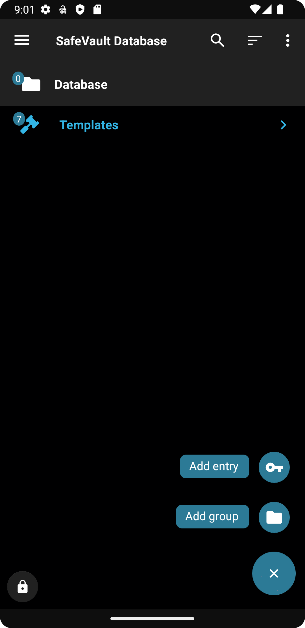
we name our database. For our implementation, the user

is known by the database name.



Thereafter, we need to set a Master Password and/or a

key file.

NOTE: Entropy denotes the lack of predictability of a

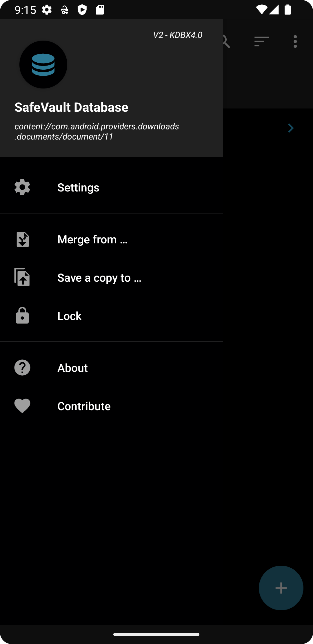
password, thus letting us know its strength. The key file

can be any file, but keep in mind that the same file will

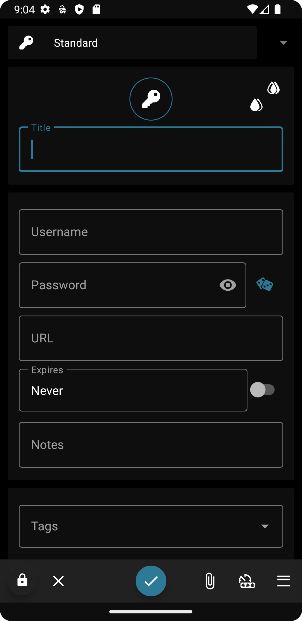
be needed to open the file again.

We finally enter our created database. We ca see a lot of

templates, which can be used for creating & storing

different kinds of data. In the bottom right corner, there’s the option to add new login credentials or add a group and put multiple login credentials into it for organizing.

On the left side hamburger menu, we can see the settings section of our database. Along with that, we can see options to add & merge existing .kdbx databases. We also have the about & contribution section of the app, where if anyone is willing to buy us a coffee for our effort.

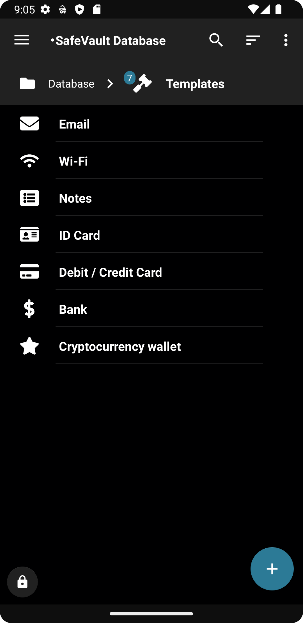


A picture containing graphical user interface

Description automatically generatedAfter clicking on add entry, we get the required fields like username and password, URL, site name, notes, and additional fields section to be filled. In the bottom right, we get buttons to add TOTP, attachments, additional fields, and save. Using the 2 drops icons we can set the foreground & background colors of the entry

After filling in all the required details, we can see the saved entry. The highlighted looks help us for an easy identification if the credential is used a lot. After this clicking on the lock icon on the bottom left will bring us back to the welcome screen of the app.

**Test & Use Cases**

Other Uses of a Password Manager

By following the above-mentioned steps, we can use our application to store data. But our apps capabilities aren’t only limited to login credentials management. Other than credentials, it can store many other different kinds of data credentials like – Email, Wi-Fi, Notes, Id-Card, Debit / Credit card, Bank & Crypto Wallet Details. All the other kinds named, have their Templates accordingly provided in our application. Users can copy / customize and use them according to their wish.

Quality-of-Service Testing

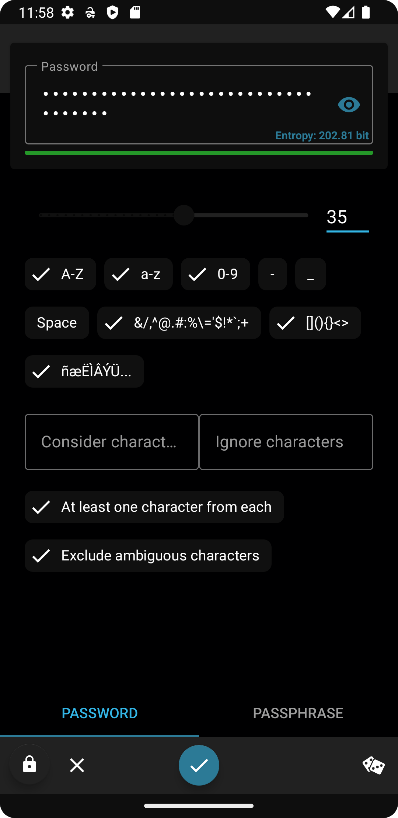
*Database Access Security*

As mentioned earlier, in our application, we have implemented a different kind of two factor authentication namely, Master Password & Key file. For Master Password, we have Entropy section which can help to determine the randomness of any password. The more the randomness, the harder to break the same. This way the user can know how difficult it is to make the Master Password difficult to ensure database safety. Moreover, Key file is a new concept, where the user can select any file within the mobile and use it as a password. Unlike other apps in the market, this doesn’t make users depend on 3rd party apps for TOTP Authentication of the Password Manager, therefore enhancing security. If these two can be stored properly by the user, there is nothing to worry about. Again, there is a popup, that states if Master Password isn’t allocated for the same.

Text

Description automatically generated*Encryption Security*

For our application, we have encryption not only for the master password but also the database too. Nothing is stored in plain text by our application & the processing of the same happens in the user’s device itself. In the case of cloud storage, before leaving the client’s device, the password gets hashed & the database encrypted. Also, it uses TLS/SSL for connecting to the cloud storage provider. Moreover, the cloud storage provider isn’t fixed, and users can get multiple options to store data accordingly. This reduces the motivation of any attacker because he/ she might get diminishing return from the attacks performed. Moreover, the encryption algorithms used are industry standards & are endorsed by international organizations, only because they are tested to be safe. Besides we can see the screenshot of out encryption selection settings present under the security section. There are a lot of metrics that can be controlled, and database access can be made much more difficult.

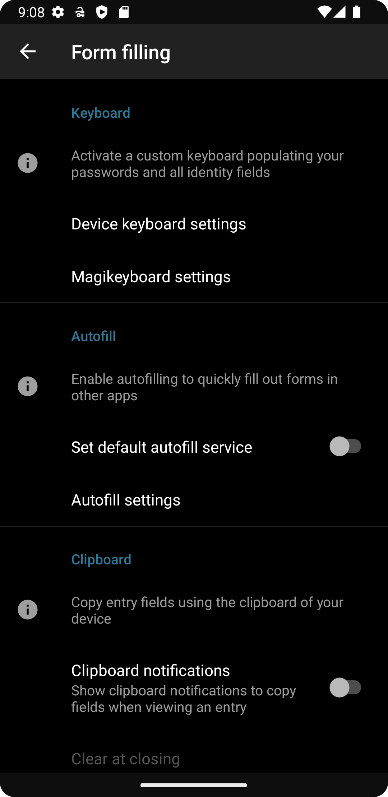


*Password Generation Security*

Under the Password Generation section, there are two kinds, namely Password & Passphrase. Some articles say that Passphrase are more secure comparatively to Password, therefore we provide support for both. Again, if we look at our implementation, we can see entropy mentioning the password strength along with the bar indicating a visual representation of the same. We also see a lot of customization options available for both the tabs. Every change made in password generation procedure affects the entropy. Now to test our entropy accuracy, copy the password and head over to any site on the internet that claims to check the password strength, and you can manually check for yourself.

*Security features Implementation*

Text

Description automatically generatedInside our app too, there are a lot of security implementations like Magi-keyboard, Screenshot Protection, Auto Timeout, History, Biometric Unlocking & Clipboard Clearing. Magi-keyboard is a simple implementation that what happens in app, stays inside app. We often see keyboards learning our usage patterns for efficient suggestions. This way it starts to learn passwords, it becomes a security risk. To prevent that enabling Magi-keyboard feature, gets us a new keyboard that can only be used inside the app thus ensuring security. Screenshot Protection helps from taking screenshots of highly critical information stored inside the app. It can be tested out by trying to take screenshot and see what toast message appears. Auto-timeout locks the database automatically after a certain period when the app is in standby, which can be checked out by keeping the app in standby. History states the changes made & what are the changes made. This helps us to trace back if anyone or the user by mistake has changed the password to a invalid one. Biometric unlocking is a simple feature that uses the phones saved fingerprint ids to unlock the database. If the phone’s Biometric Unlocking accuracy can be trusted, our app uses the same implementation & can be easily trusted. Clipboard Clearing in a simple implementation where copied password used for filling or sharing purposes gets automatically cleared after a certain duration. This feature can be tried out simply by copying any password & trying to paste it.

**Conclusion & Future Enhancements**

Currently, we are limited to mobile development, only for the Android platform. In future, if our product garners attention and a user base, it can be launched in other different platforms like Desktop, Web, CLI, Browser Extensions etc. Cross-platform support for Mac, iPhones, Linux distros too can be devised. Features like Credentials Sharing, Breach Alerts, Hardware-Key Authentication etc. can be implemented. Other than that, since the project is open source, anyone with the proper knowhow, can fork our project to implement newer innovative features and security algorithms & also improve the existing ones.

With this project, we tried to look at a very important aspect of cybersecurity, Password Management. We covered twelve features & discussed topics related to the working methodology over thirteen different password managers which we selected due to their popularity. We, through our findings, not only understood which methods are strong enough to fend hackers but also found out about features that would provide best case usability to users and would incorporate this knowledge for our implications segment of the project. We tried to create the most comprehensive Password Managing Software possible. Our Password Manager being most feature rich will appeal a larger audience base. Moreover, deploying the state-of-the-art security features, we can ensure safety of data from breaches and privacy to the highest level. Being open source allows our customers to get a knowhow about how our software works and thus makes us more trustworthy than most of the players in the existing market. Furthermore, being a mobile app, increase to reach and usage potential of the app. And this altogether would make internet safer in days to come.

**References**

1. 1Password. (n.d.). lPassword Watchtower. Retrieved April 2, 2021, from https://watchtower.1password.com/
2. 1Password. (2019, January 14). lPassword Security Design. Retrieved April 2, 2021, from https://1password.com/files/1Password-White-Paper.pdf
3. 1Password. (2020a, September 2). About Biometric Unlock Security in lPassword for Android. Retrieved April 2, 2021, from https://support.lpassword.com/android-biometric­ unlock-security/
4. 1Password. (2020b, October 27). CVE-2020-10256 For All Version of the lPassword Command­ line Tool and SCIM Bridge Released Prior to December 24, 2018. Retrieved April 2, 2021, from https://support.lpassword.com/kb/202010/
5. 1Password. (2021a, February 12). Security Audits of lPassword. Retrieved April 2, 2021, from https://support.lpassword.com/security-assessments/
6. 1Password. (2021b, February 11). Create Firewall Rules in lPassword Business. Retrieved April 2, 2021, from https://support.lpassword.com/firewall-rules/
7. 1Password. (2021c, February 9). CVE-2021-26905 For All Versions of the lPassword SCIM Bridge Released Prior to February 8, 2021. Retrieved April 2, 2021, from https://support.lpassword.com/kb/202102/
8. 8Bit Solutions LLC. (2018, November 8). Bitwarden Security Assessment Report. Retrieved April 2, 2021, from https://cdn.bitwarden.net/misc/Bitwarden%20Security%20Assessment%20Report. pdf
9. Apple. (n.d.a). Use Two-Factor Authentication for Security on Your Mac, iOS Devices, and iPadOS devices. Retrieved April 2, 2021, from https://support.apple.com/en-ca/guide/mac­ help/mchl8bd4e9c2/mac
10. Apple. (n.d.b). Accessing Keychain Items with Face ID or Touch ID. Retrieved April 2, 2021, from https://developer.apple.com/documentation/localauthentication/accessing\_keych ain- items- with- face--i-d or touch- id
11. Apple. (2021a, February 18). Keychain Data Protection. Retrieved April 2, 2021, from https://support.apple.com/en-ca/guide/security/secb0694df1a/web
12. Apple. (2021b, February 18). Secure iCloud Keychain Recovery. Retrieved April 2, 2021, from https://support.apple.com/en-ca/guide/security/secdeb202947/web
13. Bachmann, L. (2020, August 13). What are Dark Web Scans?. LastPass. Retrieved April 2, 2021, from https://blog.lastpass.com/2020/08/what-are-dark-web-scans/
14. Berba, P. {2020, May 30). How to Decrypt a LastPass Vault. lnfosec Write-ups. Retrieved April 2, 2021, from https://infosecwriteups.com/how-lastpass-decrypts-your-vault-279153350930
15. Bitwarden. (n.d.a). Security FAQs. Retrieved April 2, 2021, from https://bitwarden.com/help/article/security-faqs/
16. Bitwarden. (n.d.b). Compliance, Audits, and Certifications. Retrieved April 2, 2021, from https://bitwarden.com/help/article/is-bitwarden-audited/
17. Bitwarden, Inc. (2020, July 17). Bitwarden Network Security Assessment Report. Retrieved April 2, 2021, from https://cdn.bitwarden.com/misc/Bitwarden%20Network%20Security%20Assessment%2 0Report%20-%202020.pdf
18. Cimpanu, C. {2018, December 29). EU to Fund Bug Bounty Programs for 14 Open Source Projects Starting January 2019. ZDNet. Retrieved April 2, 2021, from https://www.zdnet.com/article/eu-to-fund-bug-bounty-programs-for-14-open-source­ projects-starting-january-2019/
19. Dashlane. (n.d.a). FAQ About Security at Dashlane. Retrieved April 2, 2021, from https://support.dashlane.com/hc/en-us/articles/360012686840-FAQ-about-security-at­ Dashlane
20. Dashlane. (n.d.b). What Are Security Alerts and Dark Web Alerts, and What to Do When I Get One. Retrieved April 2, 2021, from https://support.dashlane.com/hc/en­ us/articles/360000038180-What-are-security-alerts-and-Dark-Web-Alerts-and-what-to­ do-when-I-get-one
21. Dashlane. {2019, October 19). What if Dashlane Gets Hacked?. Retrieved April 2, 2021, from https://blog.dashlane.com/what-if-dashlane-gets-hacked-master-password/
22. Dashlane. {2020, September). Security White Paper. Retrieved April 2, 2021, from https://www.dashlane.com/download/Dashlane\_SecurityWhitePaper\_November2020.p df
23. EnPass. (n.d.a). Security. Retrieved April 2, 2021, from https://www.enpass.io/security/
24. EnPass. (n.d.b). Quick Unlock. Retrieved April 2, 2021, from https://www.enpass.io/docs/security-whitepaper-enpass/quick\_unlock.html
25. EnPass. (n.d.c). Security Vulnerability Reporting. Retrieved April 2, 2021, from https://www.enpass.io/vulnerability-reporting/
26. EnPass Team. {2017, January 11). An Update on the Reported Vulnerability of EnPass for Windows PC. EnPass. Retrieved April 2, 2021, from https://www.enpass.io/blog/enpass­ windows/an-update-on-the-reported-vulnerability-of-enpass-for-windows-pc/
27. Everis. {2016, October). KeePass Code Review Results Report. EU Free and Open Source Auditing Project. Retrieved April 2, 2021, from https://joinup.ec.europa.eu/sites/default/files/inline-files/DLV%20WP6%20-01-
28. %20KeePass%20Code%20Review%20Results%20Report\_published.pdf
29. Exploit Database. {2015, October 6). Last PassBroker 3.2.16. Retrieved April 2, 2021, from https://www.exploit-db.com/exploits/38405
30. Exploit Database. (2016, January 11). KeePass Password Safe Classic 1.29. Retrieved April 2,
31. 2021, from https://www.exploit-db.com/exploits/39216
32. Exploit Database. {2017, August 3). Dash/one - DLL Hijacking. Retrieved April 2, 2021, from https://www.exploit-db.com/exploits/44066
33. Exploit Database. (2020, September 4). Nord VPN-6.31.13.0 "nordvpn-serviceJ Unquoted Service Path. Retrieved April 2, 2021, from https://www.exploit-db.com/exploits/48790
34. FortiGuard Labs. {2018, May 14). Zero-Day Research / Fixes Pending. Fortinet. Retrieved April 2, 2021, from https://www.fortiguard.com/zeroday?type=zd&vendor=Roboform
35. Gentili, P., Shader, 5., Yip, R., and Zeng, B. {2016, May). Security Analysis of Dash/one. Retrieved April 2, 2021, from https://courses.csail.mit.edu/6.857/2016/files/25.pdf
36. Github. (2020, September 20). Security: Bitwarden Desktop App Grants RCE to Bitwarden Developers. Retrieved April 2, 2021, from https://github.com/bitwarden/desktop/issues/552
37. iClarified. {2012, August 5). How to Use 05 XJs Built-In Password Generator. Retrieved April 2, 2021, from https://www.iclarified.com/23680/how-to-use-os-xs-builtin-password­ generator
38. KeePass. (n.d.a). Security- KeePass. Retrieved April 2, 2021, from https://keepass.info/help/base/security.html
39. KeePass. (n.d.b). Awards/Ratings - KeePass. Retrieved April 2, 2021, from https://keepass.info/ratings.html
40. KeePass. (n.d.c). Password Generator- KeePass. Retrieved April 2, 2021, from https://keepass.info/help/base/pwgenerator.html
41. Keeper. (n.d.a). Keeper MSP Technical Whitepaper. Retrieved April 2, 2021, from https://www.keepersecurity.com/assets/pdf/Keeper-Managed-Service-Provider-Tech­ WhitePaper.pdf
42. Keeper. (n.d.b). Business Dark Web Monitoring. Retrieved April 2, 2021, from https://www.keepersecurity.com/breachwatch.html
43. Keeper. (2021a, March). Keeper Encryption Model. Retrieved April 2, 2021, from https://docs.keeper.io/enterprise-guide/keeper-encryption-model
44. Keeper. (2021b, March 27). How Do I Turn on Fingerprint I Touch ID Login?. Retrieved April 2, 2021, from https://help.keeper.io/article/98-how-do-i-turn-on-fingerprint-touch-id­ login
45. Kenyon, M. (2017, September 11). Serious Vulnerabilities Discovered in South Korean Parental Apps. The Citizen Lab. Retrieved April 2, 2021, from https://citizenlab.ca/2017/09/korean-parental-apps/
46. Knight, J. (2020, September 3). KeePass Gets Full Biometrics Support in the Latest KeePass2Android Update. Gadget Hacks. Retrieved April 2, 2021, from https://android.gadgethacks.com/how-to/keepass-gets-full-biometrics-support-latest­ keepass2android-update-0331354/
47. LastPass. (n.d.a). How is LastPass Safe?. Retrieved April 2, 2021, from https://support.logmeininc.com/lastpass/help/how-is-lastpass-safe-lp010089
48. LastPass. (n.d.b). LastPass Technical Whitepaper. Retrieved April 2, 2021, from https://assets.cdngetgo.com/aa/97/Sec2619b46349dc3eb3212a37b11/lastpass­ technical-whitepaper.pdf
49. LastPass. (n.d.c). Pricing by Plan. Retrieved April 2, 2021, from https://www.lastpass.com/pricing
50. LastPass. (n.d.d). IP Address: /AM Policy. Retrieved April 2, 2021, from https://www.lastpass.com/policies/ip-address
51. LastPass. (n.d.e). Dark Web Monitoring. Retrieved April 2, 2021, from https://www.lastpass.com/solutions/dark-web-monitoring
52. Leo, G., and Watson, F. (2018, November 30). EnPass Apps Security Assessment. VerSprite.
53. Retrieved April 2, 2021, from https://dl.enpass.io/docs/EnpassSecurityAssessmentReport.pdf
54. LogMeln. (n.d.). How Do I Set Up Biometrics and Mobile Account Recovery on Android for LastPass?. Retrieved April 2, 2021, from https://support.logmeininc.com/lastpass/help/how-do-i-set-up-and-use-fingerprint­ biometrics-with-lastpass-for-my-android-device
55. LogMeOnce. (n.d.a). Why Use LogMeOnce. Retrieved April 2, 2021, from https://www.logmeonce.com/why-use-logmeonce/
56. LogMeOnce. (n.d.b). Anti-Hacker. Retrieved April 2, 2021, from https://www.logmeonce.com/anti-hacker/
57. LogMeOnce. (2020, August). Vulnerability Disclosure Policy. Retrieved April 2, 2021, from https://www.logmeonce.com/vulnerability-disclosure-policy/
58. Long, H. {2020, January 6). Dash/one Review. Restore Privacy. Retrieved April 2, 2021, from https://restoreprivacy.com/password-manager/reviews/dashIane/
59. Mitre. (n.d.a). LastPass Search Results. Retrieved April 2, 2021, from https://cve.mitre.org/cgi­ bin/cvekey.cgi?keyword=LastPass
60. Mitre. (n.d.b). KeePass Search Results. Retrieved April 2, 2021, from https://cve.mitre.org/cgi­ bin/ evekey.cgi?keyword=KeePass
61. Mitre. (n.d.c). CVE-2017-11657. Retrieved April 2, 2021, from https://cve.mitre.org/cgi­ bin/ evename.cgi?name=CVE-2017-11657
62. Mitre. (n.d.c). Bitwarden Search Results. Retrieved April 2, 2021, from https://cve.mitre.org/cgi­ bin/cvekey.cgi?keyword=Bitwarden
63. Mitre. (n.d.d). NordVPN Search Results. Retrieved April 2, 2021, from https://cve.mitre.org/cgi­ bin/ evekey.cgi?keyword=nordvpn
64. Mitre. (n.d.e). Apple Keychain Search Results. Retrieved April 2, 2021, from https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Apple+Keychain
65. NordPass. (n.d.a). XChaCha20 Encryption. Retrieved April 2, 2021, from https://nordpass.com/features/xchacha20-encryption/
66. NordPass. (n.d.b). Master Password. Retrieved April 2, 2021, from https://nordpass.com/features/master-password/
67. NordPass. (n.d.c). Access with Biometric Authentication. Retrieved April 2, 2021, from https://nordpass.com/features/biometric-fingerprint/
68. NordPass. (n.d.d). Data Breach Scanner: Check Your Password Safety. Retrieved April 2, 2021, from https://nordpass.com/features/password-breach-report/
69. NordPass. (2020, February). Cure53 Security Assessment of the NordPass Complex February 2020. Retrieved April 2, 2021, from https://my.nordaccount.com/reports/nordpass­ audit-february-2020/
70. Owaida, A. (2021, March 5). Popular Password Manager in the Spotlight Over Web Trackers. We Live Security. Retrieved April 2, 2021, from https://www.welivesecurity.com/2021/03/01/popular-password-manager-spotlight­ web-trackers/
71. RoboForm. (n.d.a). RoboForm Manual for Android Mobile Application. Retrieved April 2, 2021, from https://www.roboform.com/manual-android
72. RoboForm. (n.d.b). How to Log in to RoboForm Using Windows Hello. Retrieved April 2, 2021, from https://help.roboform.com/hc/en-us/articles/360031570731-How-to-log-into­ RoboForm-using-Windows-Hello
73. RoboForm. (n.d.c). Policies I Video Tutorial Available. Retrieved April 2, 2021, from https://help.roboform.com/hc/en-us/articles/115002512592-Policies-Video-Tutorial­ Available-
74. RoboForm. (n.d.d). Key Features. Retrieved April 2, 2021, from https://www.roboform.com/key-features
75. RoboForm. {2018, October). Security Overview Whitepaper. Retrieved April 2, 2021, from https://www.roboform.com/pdf/RoboForm\_Security\_White\_Paper.pdf
76. Samsung. (n.d.). General Questions and Information About Samsung Pass. Retrieved April 2, 2021, from https://www.samsung.com/us/support/answer/ANS00066601/
77. Samsung Knox. (n.d.). Due/DAR Encryption. Retrieved April 2, 2021, from https://docs.samsungknox.com/admin/whitepaper/kpe/DualDAR.htm
78. Spearrin, K. (2017a, May 25). Have You Been Pwned?. Bitwarden. Retrieved April 2, 2021, from https://bitwarden.com/blog/post/have-you-been-pwned/
79. Spearrin, K. (2017b, October 4). Host Your Own Open Source Password Manager. Bitwarden. Retrieved April 2, 2021, from https://bitwarden.com/blog/post/host-your-own-open­ source-password-manager/
80. Support Team. (2018, May 1). How to Use Account Freeze. LogMeOnce. Retrieved April 2, 2021, from https://support.logmeonce.com/hc/en-us/articles/360003735473-How-to-Use­ Account-Freeze
81. Wagenseil, P. (2020, March 25). LastPass, 1Password, and Other Password Managers Can Be Hacked: What to Do Now. Tom's Guide. Retrieved April 2, 2021, from https://www.tomsguide.com/news/password-manager-hacks
82. Walsh, R. (2020, November 18). NordPass Review. ProPrivacy. Retrieved April 2, 2021, from https://proprivacy.com/password-manager/review/nordpass
83. Zero-Day Initiative. (2019, May 29). Samsung Knox Secure Folder Lock Screen Bypass Vulnerability. Retrieved April 2, 2021, from https://www.zerodayinitiative.com/advisories/ZDl-19-515/
84. Zoho. (n.d.a). Your Security is Our Priority. Retrieved April 2, 2021, from https://www.zoho.com/vault/vault-security.html
85. Zoho. (n.d.b). Zoho Vault-Application Security. Retrieved April 2, 2021, from https://www.zoho.com/vault/security.html
86. Zoho. (n.d.c). Steps to Configure IP Restriction in Zoho's Password Manager. Retrieved April 2, 2021, from https://www.zoho.com/vault/zoho-password-manager-ip-restriction.html
87. Oesch, Sean, and Scott Ruoti. "That Was Then, This Is Now: A Security Evaluation of Password Generation, Storage, and Autofill in Browser-Based Password Managers." USENIX Security Symposium. 2020
88. Gallagher, Elizabeth A. "Choosing the Right Password Manager." Serials Review 45.1-2 (2019): 84-87
89. Ayyagari, Ramakrishna, Jaejoo Lim, and Olger Hoxha. "Why Do Not We Use Password Managers? A Study on the Intention to Use Password Managers." Contemporary Management Research 15.4 (2019): 227-245
90. Alkaldi, Nora, and Karen Renaud. "Why do people adopt, or reject, smartphone password managers?." (2016)
91. Oesch, Sean, Anuj Gautam, and Scott Ruoti. "The Emperor's New Autofill Framework: A Security Analysis of Autofill on iOS and Android." arXiv preprint arXiv:2104.10017 (2021)
92. Das, Anupam, et al. "The tangled web of password reuse." NDSS. Vol. 14. No. 2014. 2014
93. Chiasson, Sonia, Paul C. van Oorschot, and Robert Biddle. "A Usability Study and Critique of Two Password Managers." USENIX Security Symposium. Vol. 15. 200
94. Chatterjee, Rahul, et al. "Cracking-resistant password vaults using natural language encoders." 2015 IEEE Symposium on Security and Privacy. IEEE, 2015
95. Karole, Ambarish, Nitesh Saxena, and Nicolas Christin. "A comparative usability evaluation of traditional password managers." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2010
96. Silver, David, et al. "Password managers: Attacks and defenses." 23rd {USENIX} Security Symposium ({USENIX} Security 14). 2014