JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue

JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)

An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Exploring the Landscape of Password Managers for Individual Users and Introducing an Innovative Solution

[1]Suman Garai, [2]Jasbir Singh, [3]Trishul V Biradar, [4]Umang Basantani, [5]Utsav Kejariwal, [6]Dr. Taskeen Zaidi

[1,2,3,4,5]Student, [6]Associate Professor
Department of Computer Science & IT,
Jain Deemed-to-be-University, Bengaluru, India

Abstract: With the increasing number of online services available, it is crucial for individuals to have control over their password management systems, which generate, store, and retrieve passwords, ensuring that they meet strict security standards to safeguard user information. As password management can be challenging, password managers have gained popularity, as users become more aware of the dangers of password reuse. This research evaluates the most popular password management tools on the market to assess their security and usability, examining aspects such as efficacy, convenience, and security, as well as multi-factor authentication and the program's security. The study aims to identify the best factors for creating the most secure password management system in today's world, providing insights on designing a secure and user-friendly password generation, storage, and retrieval system. By analyzing existing systems' algorithms, the research team proposes an innovative solution with a focus on security protocols, and effective password storage, and retrieval. Additionally, the research explores the potential for future improvements and advancements to the password manager. The results of this research will contribute to a more secure online environment for users and help to ensure that their personal information remains protected.

Keywords— encryption, hashing, cloud, multifactor authentication, derivation key, generation, storage, autofill.

I. INTRODUCTION

Password-based authentication is still the most used form of authentication on the web, despite its well-known challenges. Security standards often require the use of complex and unique passwords, but these types of passwords are difficult for users to remember. As a result, users often create weaker passwords that are easier to recall or use simple modifications to popular passwords. This puts them at a higher risk of being targeted by attackers. According to Lo (2016), when asked to reset a password, users tend to utilize the same password with personally identifiable information appended, such as dates of birth or nicknames. However, because a dictionary attack may quickly create probable passwords based on commonly used phrases, these personal traits are worthless in defending against individual hacking efforts (Yan et al., 2004)[1]. Additionally, human memory is limited and focuses on familiarity and repetition, making it difficult for individuals to store complex and random sequences of characters. Research has shown that people can only remember seven characters at a time, which is not enough to store the types of passwords that are considered secure in today's online world (Lo, 2016). Despite the growing number of data breaches, users still often reject security advice as it requires a significant effort, especially when the number of users affected by breaches is low. However, this situation leaves many individuals vulnerable to exploitation as the number of data breaches continues to rise.[2]

Password management systems are increasingly being used by individuals to keep their online passwords secure. These systems generate, store, and retrieve passwords for the user, thus reducing the mental load of remembering multiple passwords. A password manager stores the user's login credentials, including usernames and passwords, in an encrypted vault, which can be accessed online across multiple devices. The vault is encrypted using a master password chosen by the user. The main purpose of a password manager is to reduce password reuse, as it generates strong passwords that are then saved and easily accessible to the user. Many password managers offer a password generation pop-up window to encourage users to create strong passwords. However, there are security challenges associated with password managers. One such issue is the risk of autofill functionality being exploited by attackers to steal user credentials. In addition, password vaults may not always be secure, as unencrypted metadata or side channel leaks from encrypted data can leave users' credentials vulnerable to attack. There is also a lack of research examining the security of password generation

in password managers. While password managers do offer many benefits, including reducing password reuse, it is important to be aware of the security challenges that they may present.[3] Further research is needed to ensure the security of password generation in password managers.

II. Literature Review

In this review study, we carried out a thorough evaluation of different password management tools available in the market, taking into consideration several essential factors such as 2FA Support, Independent Security Audits, Open-Source Availability, Encryption Standards, IP Restrictions, Hosting Options, Public Bug Reporting Programs, Customizable Password Generation, Data Breach Notifications, and Vulnerability Analysis.

Our investigation of some of the most used password management solutions was useful in increasing our understanding of the security features present in these solutions. This information was then used to select a solution that meets our specific needs. The results of our investigation have been organized in an easy-to-read format, with a checkmark representing the presence of a feature and a cross representing its absence or unproven existence.[4]

In the following pages, we will delve deeper into each of the criteria mentioned above and discuss their implications for the applications. Our study aimed to provide users with a comprehensive understanding of the various password management solutions available in the market and help them make an informed decision when selecting a solution that meets their specific needs. Now, let's explore the criteria based on which password managers are classified and the implications of each criterion.

Encryption standard is a protocol used to secure data by converting it into an unreadable format. For a password manager, it is crucial to have a strong and widely trusted encryption standard, such as AES or RSA, to protect the stored passwords from unauthorized access. the strength of the encryption is determined by the key length, with a longer key length providing a stronger encryption. Different password managers may use different encryption methods and have trade-offs between security and performance.

Two-factor authentication (2FA) or multifactor authentication (MFA) involves providing two or more means of verification in order to access an account or system. The first step typically involves entering a password, while the subsequent step involves furnishing either information that the user knows, possesses, or is (such as a security question, a device, or biometric information). In the context of a password manager, 2FA can provide an extra layer of security but can also impact usability by slowing down the access process or causing inconvenience if the second factor is lost.

Biometric authentication is a type of identity verification that uses biological traits such as fingerprints or facial recognition to identify an individual. For a password manager, biometric authentication can combine something the user is with something the user knows to add an extra layer of security. However, there are concerns about biometric data being compromised and privacy issues, making it important for password managers to implement strong security measures for biometric authentication.

Password generation refers to the process of automatically creating secure, random passwords to protect online accounts. In a password manager, password generation can simplify the process of using strong passwords and the password manager can store these passwords securely. Password managers typically allow users to customize the generated password parameters, such as length and character types, and may include a password health check feature.

Breach Alerts are notifications that informs a user or organization of a data breach, which could put the stored passwords and sensitive information in the password manager at risk. In the context of a password manager, a breach alert means users should change their passwords immediately and consider updating the security measures used to protect their password manager account. The password manager should have the capability to detect potential breaches and notify users in real-time to prevent damage.

Independent Audits evaluate the security and privacy measures of an organization, service, or product by a third-party organization. In the context of password management, this would involve evaluating the encryption algorithms, data storage and transmission methods, and overall security architecture of the password manager. The results of an independent audit can provide users with an objective assessment of the security and privacy features of a password manager and help organizations evaluate the security of their password management systems.

A public bug bounty program invites security researchers and ethical hackers to identify and report vulnerabilities in a company's software in exchange for a reward. This type of program can help improve the security and privacy of user data stored in a password manager by encouraging third-party experts to report any vulnerabilities that could be exploited by malicious actors. A public bug bounty program can also demonstrate the company's commitment to the security and privacy of their software, increasing customer confidence.

Past vulnerabilities in password managers refer to security flaws or weaknesses discovered in earlier versions of the software. This can lead to the loss of sensitive information such as login credentials and personal data. Present vulnerabilities refer to security flaws or weaknesses currently present in the software, and can be discovered through various means, including security audits and reports from users. The implications of these vulnerabilities can result in the loss of sensitive information and identity theft. It is important for users to stay informed about the latest vulnerabilities in their password manager and take action to address them promptly.

The storage methods used by password managers can also impact the security and privacy of user data. Some password managers store data on a server controlled by the company, while others allow users to store their data locally on their own devices. Storing data on a server controlled by the company can make it easier for the company to maintain and secure the data, but it can also make the data more vulnerable to cyber-attacks and data breaches. On the other hand, storing data locally on the user's device can provide more control over the security of the data, but it can also make it more difficult for the user to access their data from multiple devices.

Another aspect to consider is whether the password manager is open-source or not. Open-source software is code that is publicly available for anyone to inspect, modify, or distribute. Open-source password managers can provide a higher level of transparency, as the code can be inspected and reviewed by security experts. However, open-source password managers can also be more vulnerable to security issues, as the code is publicly available and can be more easily exploited by malicious actors. On the other hand, proprietary

password managers, which have their source code proprietary and only accessible to the company, can provide a higher level of security, as the code is not publicly available and can be more easily protected.

Choosing a password manager involves considering a number of important factors, which includes the results of encryption method used, MFA/ Biometric Authentication presence, strong password generation capabilities, breach alerts, independent audits, the presence of public bug bounty programs, the history of past and present vulnerabilities, the availability of IP whitelisting and tracking features, the storage methods used, and whether the software is open-source or proprietary. By taking the time to consider these factors and make an informed decision, users can help to ensure the security and privacy of their sensitive information and protect themselves from potential data breaches and other security risks.

Table 2.1: Comparison of different Popular Password Manager Features available for Users.

Tool	2FA/MFA Support	Independent Audit	Open Source	Encryption Standard	Self-Hosting Support	IP Whitelisting	Public Bug Bounties	Class	Password Generation Options	Mobile App Biometrics	Breach Alerts	Present / Past Vulnerabilities	Privacy Officer Response
1Password	✓	✓	✗	AES256	✗	✗	✓	Web, Mobile	Length of 0-100 with letters, numbers, and symbols	✓	✓	✓	1 Hour
BitWarden	✓	✓	✓	AES256	✓	✗	✓	Desktop, Web, Mobile	Length of 5-128 with letters, numbers, and symbols	✓	✓	✓	12 Hours
NordPass	✓	✓	✗	XChaCha20	✓	✗	✗	Desktop, Web, Mobile	Length of 8-60 with letters, numbers, and symbols	✓	✗	✗	16 Hours
Zoho Vault	✓	✗	✗	AES256	✗	✗	✓	Web, Mobile	Length of 4-100 with letters, numbers, and symbols	✓	✗	✗	9 Hours
LastPass	✓	✗	✗	AES256	✗	✗	✓	Web, Mobile	Length of 0-99 with letters, numbers, and symbols	✓	✗	✓	16 Hours
KeePass	✗	✓	✓	AES TwoFish XChaCha	✓	✗	✗	Desktop, Mobile	Variable length with letters, numbers, and symbols	✓	✗	✓	✗
Keeper	✓	✗	✗	AES256	✗	✗	✓	Web, Mobile	Variable length with letters, numbers, and symbols	✓	✓	✓	½ Hour
Enpass	✗	✓	✗	AES256	✗	✗	✗	Web, Mobile	Variable length with letters, numbers, and symbols	✓	✓	✓	5 Days
iCloud Keychain	✓	✗	✓	AES256	✗	✗	✓	Mobile	Variable length with letters, numbers, and symbols	✓	✗	✓	2 Weeks
Dashlane	✓	✗	✗	AES256	✗	✗	✓	Web, Mobile	Length of 4-40 with letters, numbers, and symbols	✓	✗	✓	✗
LogMeOnce	✓	✗	✗	AES256	✓	✗	✓	Desktop, Web, Mobile	Length of 6-50 with letters, numbers, and symbols	✓	✗	✗	✗
RoboForm	✓	✗	✗	AES256	✓	✗	✗	Desktop, Web, Mobile	Variable length with letters, numbers, and symbols	✓	✗	✓	✗
Samsung Pass	✓	✗	✗	AES256	✗	✗	✓	Mobile	✗	✓	✗	✓	✗

1Password is a password management solution that employs AES-256 encryption for safeguarding user data and 2-Secret Key Derivation (2SKD) to ensure secure access to the account. A master password is only one of the two required secrets, while the other is a randomly generated, cryptographic string.[5] 1Password supports multi-factor authentication (MFA) through Authy or Microsoft Authenticator and the Android version of the app supports biometric authentication. The password generator generates random passwords with an option to select the length, letters, numbers, symbols, and characters used. The password generator can also generate memorable passwords and pins. It features "Watchtower," which checks if any of the user's passwords have been detected in password dumps.[6] The company has undergone multiple security audits, assessments, and penetration tests, with the most recent audit by Onica showing no high-risk issues. However, the most recent penetration test by Cure53 showed two medium-risk threats and that the 1Password Vault was vulnerable to compromise.[7] 1Password offers a public bug bounty program for vulnerabilities in its website, sign-up procedure, authentication, and in-app features, however, the program does not compensate for bugs resulting from scheduled infrastructure changes, header issues for session management, or exploits that necessitate elevated access like DDoS/DoS attacks. As of now, 1Password has two known CVEs, one for a vulnerability in the SCIM Bridge platform that allowed for viewing of the TLS private key for internet connections[8] and another for the use of insecure RNG to generate keys.[9] The capability of IP whitelisting is available in 1Password for businesses, however, it is not a feature in the standard version of the password management tool.

Bitwarden is an open-source password manager that provides secure storage for all vault-related information. The data is safeguarded through the use of AES-256 encryption, and the AES keys are generated from the master password using the SHA256 algorithm. The encryption key for unlocking the data is kept in the system only when the Bitwarden app is accessible. The platform offers multifactor authentication options for both standard and premium members, which include an authenticator app, email, Duo Security, VubiKey, or FIDO U2F.[10] Both the Android and iOS apps support biometrics, and users can configure the feature in the settings. Bitwarden has a password generator that can generate passwords with lengths ranging from five to 128 characters and allows users to customize the style and characteristics of the password. Bitwarden offers a "Data Breach Report" function, which uses HavelBeenPwned[11] to identify password dumps and is available on the free version for checking one password at a time. Bitwarden undergoes regular independent audits, and results are posted in a public security assessment report.[12] Bitwarden operates a bug bounty program through the HackerOne platform, covering any vulnerabilities that may impact its products. Bitwarden has two previous CVEs, including a server-side request forgery issue from 2020 and a potential KDF vulnerability in 2019.[13] There is also a possible remote code execution vulnerability through the auto-update feature.[14] The platform allows for self-hosting through local storage or Docker and hosts all its code on Github.

NordPass is a password manager that uses XChaCha20 encryption[15] to protect the user's passwords. It has a zero-knowledge architecture, meaning that the encryption keys are not stored on the developer's infrastructure, but instead, kept locally on the user's device.[16] The master password is not stored anywhere and can only be unlocked by the user. NordPass offers a range of MFA options, including Google Authenticator, Duo, and Authy, and also supports fingerprint authentication on its mobile app. The

password manager also has a password generator that can create a password between 8 to 60 characters, with the option to include letters, numbers, and symbols, or avoid ambiguous characters. The premium version of NordPass offers a "Data Breach Scanner" feature that checks password dumps against the user's stored information.[17] The results of an independent audit by Cure53 are available for viewing by Nord account holders.[18] NordVPN, the company behind NordPass, operates a bug bounty program that does not extend to the NordPass infrastructure. At present, there have been no reported vulnerabilities or exploits affecting NordPass. However, NordVPN has previously been the target of CVEs like public exploits that involve code execution and elevated local privilege.[19] NordPass allows for local hosting through a password database and does not require a Nord account to use.

Zoho Vault employs the AES-256 encryption for all confidential information and does not keep the master password in its storage. The data is sent over the internet in AES encrypted form and is protected by TLS with strong ciphers for all connections.[20] Zoho Vault provides an added layer of security with its multi-factor authentication options including voice call, text message, Zoho OneAuth, Google Authenticator and Yubikey. The mobile app of Zoho Vault also supports biometric authentication through Swift Login setting.[21] The password generator in Zoho Vault creates passwords of length four to 100 characters, with options for numbers, special characters, letters, starting with a letter and mixed case letters. Zoho has a self-hosted bug bounty program that covers all Zoho-branded products and applications, including Zoho Vault.[22] IP whitelisting[23] is supported in the standard, professional, and enterprise versions, but not in the free version.

LastPass is a password manager that aims to secure users' online accounts and personal information. The platform uses a master password to generate AES-256 keys, which are then hashed multiple times using the PBKDF2 SHA256 algorithm. After this process, the master password is further hashed and stored as an authentication hash.[24] LastPass provides multi-factor authentication in its free plan with a variety of options, such as Duo Security, Google Authenticator, LastPass authenticator, Microsoft Authenticator, Grid, and Toopher. The mobile app of LastPass offers biometric authentication, which can be set up right after installation. The user is prompted to scan their fingerprint to enable biometrics in the security settings. The platform also offers biometric account recovery.[25] The password generator offered by LastPass allows users to choose the length of the password, which ranges from 0-99 characters and there are three options available, "easy to say", "easy to read, and "all characters". LastPass also offers Dark Web Monitoring as a feature for its premium and above versions. This feature alerts users via email if their email addresses are detected in password dumps from Enzoic.[26] The platform does not have an independent audit and has a public bug bounty program that covers the product, browser extensions, desktop applications, and mobile applications. However, there are certain areas of security that the platform does not cover, such as two-factor authentication mobile apps, information leaked through memory dumps, desktop applications compromised by malicious software or browser extensions, and man-in-the-middle (MITM) attacks.[27] LastPass has six past CVEs,[28] two of which are from 2020 and are disputed due to the vulnerability relying on a jailbroken device. LastPass, as a paid password manager, does not support IP whitelisting natively but offers it through its paid 'Identity' service. The Android application of LastPass has seven tracking features, which raises privacy concerns.

KeePass is an open-source, free-ware password manager that uses AES, TwoFish, and ChaCha20 algorithms to encrypt usernames, passwords, and notes, and SHA256 to hash the master password for authentication.[29] The Android version of KeePass, although not officially supported, provides the option of biometric authentication.[30] KeePass features a password generator with user-defined length and an extensive symbol set that includes letters, numbers, and special characters, as well as the ability to create passwords based on customizable patterns. However, it does not have a feature for breach alerts. However, a security audit of KeePass was carried out in 2016 by the Free and Open-Source Software Auditing project of the European Commission, with the full results made public. The audit discovered five medium-level issues, but no critical ones. It is unknown if the European Union's plan to fund KeePass for a bug bounty program in January 2019 was carried out, as there is no recent information available on the matter.[31] KeePass has seven past CVEs,[32] with two from 2020 that could lead to data reading or modification, and two public denial of service exploits. It also provides the option for users to host the password manager themselves or use a hosting service such as Dropbox.

Keeper is a paid password management solution that utilizes AES-256 encryption to secure passwords and other data. This encryption is performed at the device level before being transmitted to the servers. The platform uses a zero-knowledge architecture, meaning the master password and data are not stored in plaintext form, and AES keys are generated from the user's master password.[33] Keeper offers multi-factor authentication through various methods like text messages, Microsoft/Google Authenticator, Duo Security, Yubikey, etc. and also biometric authentication for iOS (Touch/Face ID)[34] and Android (fingerprint). Additionally, the app provides a password generator for generating random passwords with letters, numbers, and symbols. The BreachWatch feature for breach alerts is not included in the free version but is available as a paid business add-on.[35] Keeper claims to undergo regular audits by firms such as NCC Group, Secarma, Rhino Security, and Cybertest, and has enacted a bug bounty program through Bugcrowd. However, no present or past vulnerabilities were noticed during research. Whitelisting specific IP addresses is not available in the free version of Keeper, but it can be established using Active Directory in the business version of the platform.

EnPass secures passwords and data with AES-256 encryption, which is applied locally on the device. The encryption key is generated from the master password, which goes through 100,000 rounds of PBKDF2-HMAC-SHA512.[36] The password manager supports biometric authentication through its mobile app[37] and has a comprehensive password generator that allows users to generate passwords with specific criteria, such as length and number of special characters, numbers, and uppercase letters. EnPass includes a feature that allows it to compare the passwords saved in its vault against the HaveIBeenPwned database and notify users if any of their passwords have been compromised.[38] EnPass has undergone a security audit by VerSprite in 2018, and the results showed a medium overall risk impact, with two vulnerabilities identified. One vulnerability was in the Windows desktop application and another was found in the Android app which disclosed the unencrypted master password. Additionally, EnPass has been affected by two previous security issues, including a local file inclusion attack in 2017 and a code injection vulnerability in 2020.[39]

The Keychain password manager uses two separate sets of AES-256 encryption keys, the table key and the per-row key, to secure its data.[40] This technology has been developed by Apple and is open-source. Keychain uses a combination of AES-256 encryption and Apple's "Secure Enclave" to secure the data stored in it. The table key is cached for improved performance, while the per-row

key is protected by the "Secure Enclave."[41] Keychain requires MFA with Apple ID and supports touch or face ID for authentication.[42] It has a password generator with adjustable length and options. There is no mention of breach alerts or a public bug bounty program, and no independent security audits have been conducted. Keychain has had several CVEs related to obtaining items, with the latest one reported in 2018.[43] The source code for Keychain is available on Apple's open-source subdomain, Apple Public Source License (APSL), for developers to review and contribute to its development.

Dashlane uses AES-256 encryption for securing passwords and employs Argon2D to generate the AES keys. It does not store the master password and deals only with AES encrypted data. The desktop application provides the option to enable multi-factor authentication, and biometric authentication can be configured through security settings.[44] Dashlane's password generator offers a length range of 4 to 40 characters and includes letters, numbers, and symbols. The premium versions of Dashlane offer "Dark Web Monitoring" which monitors up to five email addresses for password breaches.[45] Dashlane claims to have regular security audits but there is very less publicly available information about them.[46] They have a public bug bounty program hosted on Hackerone that covers autofill/autologin, the website, API endpoints, client applications, and standalone extensions. In the past, Dashlane had a vulnerability related to DLL hijacking,[47] but it has since been resolved. Dashlane is a proprietary software, but parts of its code can be accessed through its active Github repository.

LogMeOnce is a paid password manager that claims to use AES-256 encryption to secure its users' passwords. However, detailed information about its encryption method is not readily available and is only available in the administrator's package. LogMeOnce offers a range of multi-factor authentication options, including voice call, Selfie 2FA, TOTP, SMS, email, X.509 certificate, USB flash drive, and security key. It also has the option for biometric authentication via fingerprint scanning. The password generator creates passwords with a length ranging from six to fifty characters, composed of letters, numbers, and symbols. LogMeOnce's password generator includes an estimation of the time needed to crack the password after hashing, giving users a clear idea of its strength.[48] Breach alerts such as monitoring for leaked passwords, dark web monitoring, and anti-theft protection are available but can only be purchased. LogMeOnce has a public bug bounty program with lower reward amounts compared to other password managers, with a maximum reward of fifty dollars.[49] The account can be frozen and access can be blocked from other IPs,[50] and storage modes can be changed between local and cloud storage.

RoboForm uses AES-256 encryption to secure passwords and data, and the encryption keys are generated from the master password and managed by RoboForm. [51] This password manager requires a paid subscription for access to its features. RoboForm offers MFA options through text message, email, or Google Authenticator, and biometric authentication through its mobile app and Windows Hello. The password generator in RoboForm can generate passwords with a variable length, controlled by a text box, and includes letters, numbers, and symbols.[52] No information was found regarding breach alerts or a public bug bounty program, and researchers have found vulnerabilities in RoboForm, including PIN bruteforcing and clipboard data theft.[53] RoboForm also offers IP whitelisting in its business version and allows local-only storage in the desktop and mobile applications.[54]

Samsung Pass protects sensitive data with Samsung Knox's encryption standard that's referred to as "military-grade." This system uses "Dual Data-at-Rest" to encrypt the data twice, utilizing two different keys, one being AES-256 encryption.[55] The inner encryption layer can be customized with a third-party cryptographic module. The National Information Assurance Partnership (NIAP) has certified Samsung Knox's encryption framework.[56] Samsung has a bug bounty program for its mobile devices, including Knox, but some vulnerabilities are excluded and the definition of "low probability of exploitation" isn't specified. In 2019, an exploit was found that allowed an attacker with physical access to retrieve sensitive data from the Samsung Knox secure folder.[57]

III. Working Methodology of Existing Solutions

It's not just the features that are important in a password manager, but also how it operates in the background. The process of storing and transferring user credentials is crucial in ensuring security and protecting against potential threats. During the transfer of data to the cloud, a Man-In-The-Middle attack can occur, and the lack of encryption in plaintext storage leaves user credentials open to theft. Additionally, unsecured apps and Java scripts on devices can be used to access password manager databases.

To better understand the security measures used by password managers, we will delve deeper into the workings of some of the most reliable password managers that have established their reliability over time.

BitWarden

Bitwarden utilizes PBKDF2 to enhance the security of a user's Master Password by combining it with a salt derived from their email address to generate a 256-bit Master Key. This key undergoes additional strengthening to 512 bits through HKDF. Unlike other password managers, Bitwarden does not store or send the Master Key to its servers. Instead, it transmits an AES-256 encrypted Protected Symmetric Key using the Stretched Master Key and an Initialization Vector. Additionally, when creating an organization, a RSA key pair and an encrypted Organization Symmetric Key with the user's Symmetric Key are generated. For authentication, a hash of the Master Password is sent to the servers during account creation and login. [10]

When logging in, Bitwarden requires the user to input their Email Address and Master Password. The latter is then transformed with PBKDF2 and the salt of the email to create the 256-bit Master Key. The hash of this key is sent to the server to authenticate the user, then further strengthened to 512 bits with HKDF. All decryption and Vault Item retrieval takes place on the Bitwarden client using the decrypted Protected Symmetric Key and the Symmetric Key, ensuring the Master Password or Stretched Master Key is never stored or transmitted to Bitwarden's servers.[10]

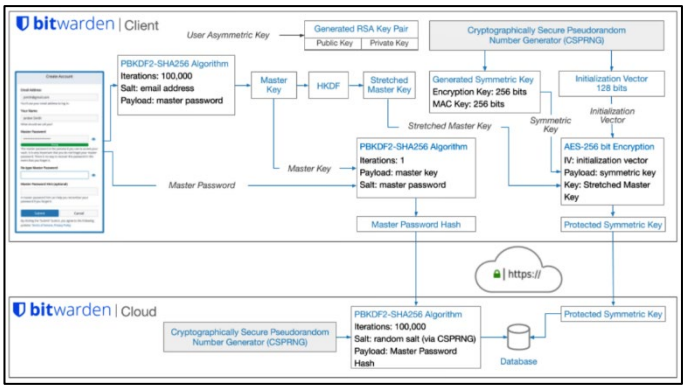

Figure3.1: BitWarden Encryption & Decryption Procedure

LastPass

LastPass employs advanced security measures to protect its users' data. When logging into the password manager, one must enter their username and a Master Password which is utilized to verify the account and unlock saved credentials. The data stored in the vault is encrypted locally and not on LastPass servers, and sensitive information is transmitted securely to prevent unauthorized access. The encryption utilized by LastPass is AES 256-bit, which is a military-grade encryption in Cipher Block Chaining (CBC) mode and is generated with a key created from each user's Master Password. The Master Password is transformed into a hash using PBKDF2-SHA256 and Scrypt and is then sent to the server for verification, but not in its original form. The encryption key and Master Password never leave the user's device, making it impossible for LastPass to access or reverse them.[24]

The encryption key and login hash are created on the user's device through thousands of rounds of PBKDF2 SHA-256, making it extremely difficult for a computer to hack the Master Password. The login hash is then transmitted to the server for verification. LastPass implements PBKDF2 server-side as well to ensure maximum protection of both the locally stored and server-stored data.[24]

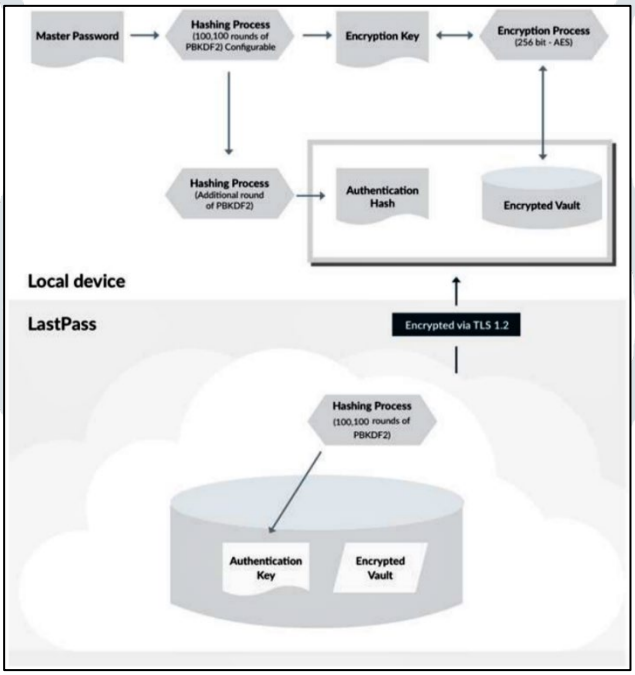

Figure3.2: LastPass Password Storage & Synchronization Procedure

In summary, LastPass employs robust encryption techniques and numerous rounds of PBKDF2 SHA-256 to create a secure login hash, making it almost impossible for anyone to access the sensitive information stored in the user's account. However, the high number of rounds utilized for encryption may cause slower login times for users using outdated browsers such as Internet Explorer.

1Password

The security of 1Password lies not only in the encryption algorithms and protocols used, but also in the proper generation, management, and protection of the keys used for encryption. 1Password secures its encryption by using strong key management practices. The encryption algorithm used is RSA-OAEP, with a 2048-bit modulus and a public exponent of 65537. To generate the keys, the client employs Cryptographically Secure Pseudo-Random Number Generators (CSPRNGs). These keys are further secured through two-secret key derivation (2SKD), which helps prevent brute-force attacks.[5]

The keys are derived from the user's account password and a Secret Key. This process starts by normalizing the account password and preparing a salt using the lowercase version of the user's email address. The password is then processed through the PBKDF2-HMAC-SHA256 hash function, which has been chosen for its efficiency across various clients. However, the slow performance in JavaScript in web browsers limits the use of more advanced password hashing schemes.[5]

The Secret Key is combined with the account password to generate an intermediate key, and then the authentication key is derived in a similar manner but with a different salt for the PBKDF2 rounds. The resulting 32-byte key is converted for use with the SRP protocol.[5]

To create a secure authentication process, a separate salt is utilized for the PBKDF2 rounds when deriving the authentication key, which differs from the method used for generating the Account Unlock Key (AUK). This results in a 32-byte key that is transformed into a BigNum format for use with SRP. Different tools are utilized depending on the platform - JSBN library in browsers and OpenSSL on other platforms. To derive the keys, 200,000 rounds of PBKDF2 are required, while an attacker only needs 100,000 rounds per attempt, leading to a minor advantage. Nonetheless, the sequence is infrequently performed, and the SRP-x is often stored locally or encrypted with the AUK. The client only needs to go through the derivation process during the initial sign-up or when enrolling a new client.[5]

During the process of adding a new device, the user supplies the new device with both the "add-device" link and their account password. The "add-device" link is generated by an already enrolled device and contains information such as the team domain name, user's email, and a secret key. The link uses the "onepassword:" format, with fields for email, server, and key included in the query string. The new device does not have its own unique salts or key derivation parameters, so it must request them from the server. Upon successful authentication, the device obtains the encrypted personal key set, including a private key, public key, and symmetric key used to encrypt the private key, all encrypted using the AUK and specific parameters and a salt.[5]

NordPass

NordPass Business utilizes encryption technology to guarantee the protection of user data. All data belongs to the organization, and if an employee leaves, the data remains within the organization through the use of public-key cryptography. Each user has a unique key pair, and their private key is encrypted using a secret key on their device. The encryption process utilizes Argon2id for the derivation of the Master Key, XChaCha20-Poly1305-IETF for secret-key cryptography, and X25519-XSalsa20-Poly1305 for public-key cryptography.[16]

The private key is only stored in plain text on the user's device temporarily and is encrypted using the Master Key, which is generated from the Master Password and a unique salt. The app stores the unencrypted private key in secure memory while the app is unlocked and deletes it when the app is locked. Every item in the app has both metadata and secret data, allowing for more precise control of permissions.[16]

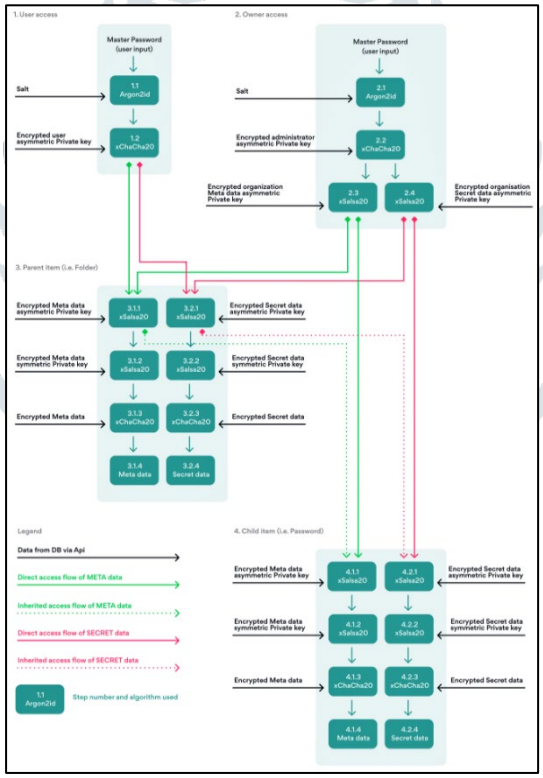

Figure3.3: NordPass Password Storage & Synchronization Procedure

Every item can be accessed through two methods: Direct access flow and the common flow. The Direct access flow is utilized when an item is shared with a user. The user is asked to enter their Master Password, which, together with the unique per-user cryptographic 16-byte salt, is used to derive the Master Key through the Argon2id key derivation function. The Master Key is then used to decrypt the user's private key.[16]

The user's encrypted private key is then decrypted locally on their device using the XChaCha20-Poly1305-IETF algorithm and the Master Key as the decryption key. Since every item has both metadata and secret data, the user's private key is used to decrypt either the item's metadata private key or secret data private key, depending on the permissions granted to the user.[16]

The private key of the item's asymmetric key pair is then used to decrypt the item's symmetric key through the xSalsa20 algorithm. Finally, the symmetric key is used to decrypt either the item's metadata or secret data using the XChaCha20-Poly1305-IETF algorithm.[16]

Roboform

RoboForm requires new users to establish a Master Password, which serves as the solitary password required to access saved information both locally and online. To secure the data, the company implements two distinct cryptographic functions to create the symmetrical key used for local encryption/decryption and server-side password protection. These functions each utilize a different, user-specific "salt" that is randomly generated.[51]

To produce the AES encryption key, RoboForm employs the PBKDF2 algorithm and the SHA-256 hash function, as well as a long random salt of 32 bytes. This process only occurs on the user's device, as the company does not perform any encryption or decryption on the server. Furthermore, user information is never transmitted to the server in an unencrypted format, with all communication between RoboForm clients and the server conducted through secure channels.[51]

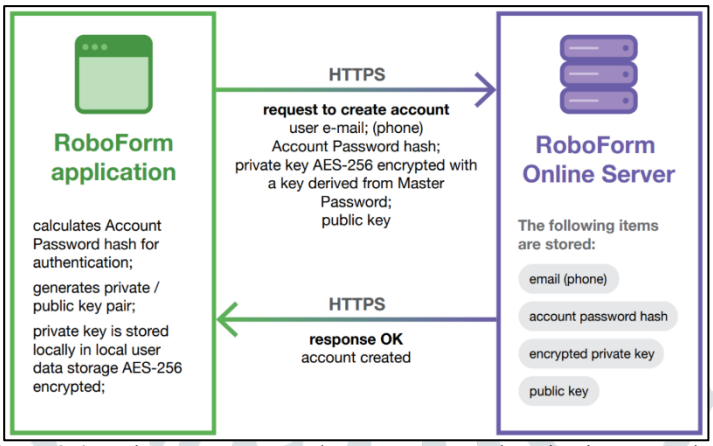

Figure3.4: RoboForm Password Storage & Synchronization Procedure

The Master Password created during the account creation process is the only password that the user is required to remember, and it serves as the key to unlocking their RoboForm data. To enhance security, two separate cryptographic functions are used to generate the symmetrical key for local encryption/decryption and server-side password protection, each with its own unique user-specific "salt".[51]

The AES encryption key is generated by utilizing the PBKDF2 (Password-Based Key Derivation Function 2) algorithm and the SHA-256 hash function, along with a long random salt. The number of iterations within PBKDF2 offers protection against brute force and dictionary attacks, however, it can also slow down the algorithm, particularly on slower devices or applications. To counteract this, it is recommended to increase the length of the password instead of the number of iterations.[51]

Only the password hash derived from the Master Password is shared with the RoboForm server, and it is impossible to recover the Master Password or AES-256 key from this hash. This added level of security ensures that the user's data remains confidential and protected.[51]

Keeper

Keeper Security utilizes a Zero-Knowledge Architecture, a security system known for its superior privacy and protection features. It is based on several key principles to maintain the security of user data. Firstly, the data is only decrypted and encrypted on the user's device and never on the server. This means that the application never saves the data in a readable format and the server never receives unencrypted information. Furthermore, neither Keeper employees nor any third parties have access to the unencrypted data. The keys used to encrypt and decrypt data are derived from the user's Master Password.[58]

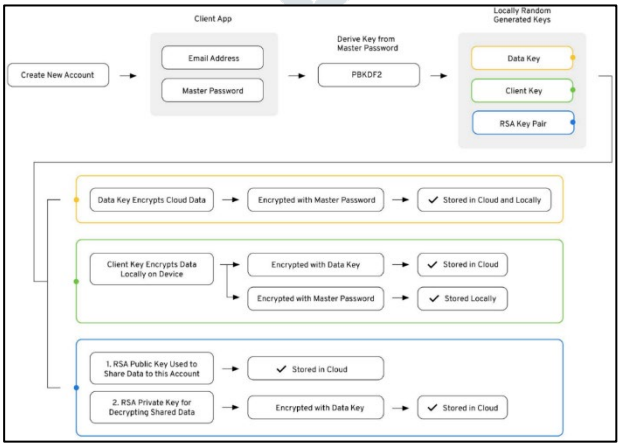

Figure3.5: Keeper Encryption Model

In addition, Keeper employs multiple layers of encryption, providing control access at the user, group, and administrator levels. Sharing of information is also secured with the use of Public Key Cryptography, which ensures safe key distribution. Keeper is committed to maintaining the highest levels of security and privacy and has received certifications such as SOC 2, ISO 27001, and

is compliant with the EU-U.S. Privacy Shield program. Regular audits are also conducted to guarantee the continuous development of secure software.[58]

To use Keeper, a unique Master Password must be chosen by the user. The Zero-Knowledge Architecture guarantees that no one, including Keeper employees and administrators, has access to this password. The administrator can set guidelines for the Master Password and in the event of a lost password, the user can recover their account through a secure process that involves a security question, email verification, and two-factor authentication.[58]

Keeper uses encryption to ensure the safety of user data, which is why it is recommended by the National Institute of Standards and Technology and the European Union's General Data Protection Regulations. The company implements symmetric encryption to store passwords in an encrypted form in a digital vault. The encryption key to access the vault is derived from the user's Master Password and all encryption keys, such as the Data Key, RSA Private Key, Record Keys, and Folder Keys, are unique to the user and encrypted for extra security.[58]

To protect the data, Keeper employs the strongest forms of encryption, including 256-bit AES and PBKDF2 for key derivation. The application uses multiple layers of encryption, including at the record, folder, and team levels. This allows for records to be shared among authorized users without risking unauthorized access. The encrypted vault is stored in the cloud for synchronization and can also be used offline, but the Keeper Administrator can restrict offline access. Data in transit is protected with 256-bit TLS/SSL encryption, further secured by Key Pinning and encryption layers to prevent man-in-the-middle attacks.[58]

Dashlane

Dashlane takes protecting user data very seriously and employs four distinct secrets to ensure the security of this information. The first of these secrets is the User Master Password, which is not stored on Dashlane servers or any of its affiliates, including hashes. By default, the Master Password is not saved on the device, but instead is used to encrypt and decrypt local data files. However, if the user wants to, they can opt to save the Master Password locally through the "Remember my Master Password" feature. The Master Password is never transmitted online.[44]

In certain circumstances, an Intermediate Key encrypted with the Master Password is used for local storage. A unique User Device Key is generated for each device that the user enables, automatically. This key is then used for authentication purposes. A local secret key is also generated, which is used to secure communication between the Dashlane application and browser plugins. The key exchange is completed through local visual pairing and Diffie-Hellman.[44]

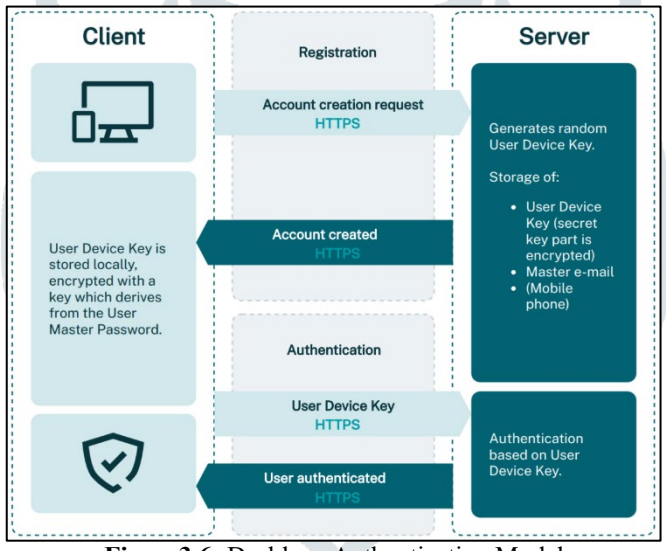

Figure3.6: Dashlane Authentication Model

Access to the user's personal information is only possible with the use of the User Master Password. This password generates a 256-bit AES key which is used to encrypt and decrypt the user's data on their device. The encryption process utilizes various libraries, such as OpenSSL or Web Crypto API, depending on the device being used. When the user inputs their Master Password into Dashlane, the data is decrypted and stored in memory. However, to maintain security, the data is encrypted using AES before being transmitted through named pipes or web sockets. The encryption process includes a random 16-byte initialization vector and a salt that is written in the AES file.[44]

All communications between the Dashlane app and servers are protected by HTTPS and SSL/TLS. The HTTPS connections are implemented on the client side using OpenSSL and on the server side with a DigiCert High Assurance CA-3 certificate. The best cipher and hash algorithm are negotiated between the client and server, and the server sends a digital certificate that the client verifies with a Certificate Authority. A symmetric key is then generated to encrypt and decrypt data. Communication between the Dashlane browser plugin and application is also encrypted using AES 256 encryption with the OpenSSL library, which includes a 32-byte salt, a randomly chosen 16-byte IV, and the salt included in the encryption process.[44]

Zoho Vault

When establishing a Zoho Vault account, users must create a master password that will serve as their encryption key. The master password must be a minimum of 8 characters long, and users receive instant feedback on its strength. This password is kept confidential by the user and never stored on Zoho's servers. Zoho Vault uses the master password to generate the Key Encryption Key (KEK) through many iterations of PBKDF2 with HMAC-SHA256 Key Derivation Function, using a random salt value.[21]

To ensure the security of sensitive user data, Zoho Vault employs a host-proof hosting method. This means that all data encryption and decryption take place within the user's browser on the client side. The data is encrypted using AES-256 encryption on the client side, transmitted securely over HTTPS, and stored on Zoho's servers in encrypted form. The master password set by the user acts as the encryption key, and it is never stored on Zoho's servers. When a user wants to access their data, the encrypted data is retrieved over HTTPS. When adding, deleting, or modifying data, Zoho Vault encrypts the data on the client side before sending it to Zoho's servers. Zoho's servers only hold encrypted data that can only be decrypted using the user's master password and a unique salt value. This means that even if someone were to access Zoho's servers, they would not be able to view the data in its original form.[21]

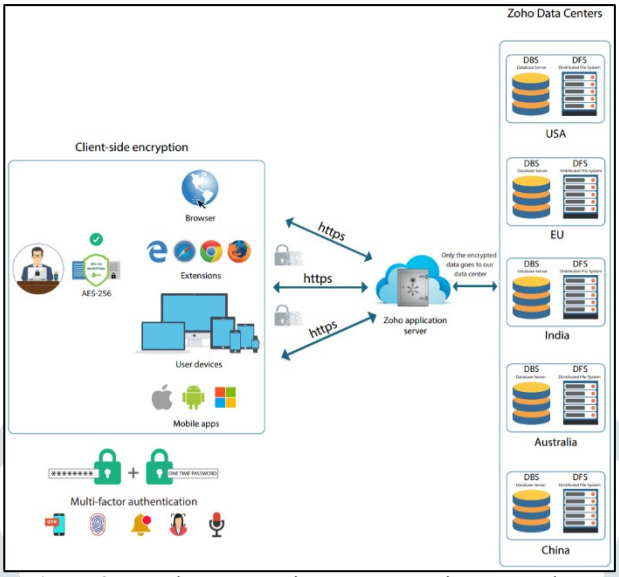

Figure3.7: Zoho Encryption & Decryption Procedure

The authentication process for each user involves several steps. Requests for access from web, browser extension, or mobile apps are directed to the Zoho Accounts login page. The login information is sent to Zoho's server for verification. If the user is authenticated, a cookie is set for the current browser session, and access to the Vault is granted. The Zoho Accounts agent on the application server then checks the cookie information with the Zoho Accounts server in the background.[21]

IV. Result & Analysis

Password managers are a vital resource for safeguarding online accounts from cyber threats. These tools generate strong, random passwords that enhance security against hacking attempts. Some password managers give users the choice of either retaining their current non-random passwords or generating new ones during the sign-up process. However, the efficacy of passwords decreases when the symbol set used is limited. Of the available options, KeePassXC provides the most extensive symbol set, including both normal and extended ASCII characters (excluding spaces). Other popular password managers like Dashlane, 1Password, etc. support only the standard ASCII symbol set, with a restricted number of characters.

Password storage is the second phase of the password manager lifecycle. Both app-based and extension-based password managers use AES-256 encryption to secure their databases. This advanced encryption technique provides robust protection for metadata stored in the password manager. All metadata is encrypted in all password managers, including KeePassXC.

Password managers employ various techniques to store and automatically fill in passwords. For instance, app-based password managers can copy cloud data locally and encrypt it using a master password. However, the autofill feature may pose a security risk if the password manager automatically fills in the password without seeking user interaction. To mitigate this risk, KeePassXC, Bitwarden, and RoboForm all require user interaction by default before autofill occurs. Autofilling passwords within iframes is also a potential security hazard, as attackers can acquire sensitive information through clickjacking or cross-domain iframe autofill.

The accepted method for storing password information is AES-256 encryption with PBKDF2 for transport to the cloud. All OS-based mobile autofill frameworks demand user interaction before autofill, providing a secure and accessible way to recall password data. iOS password autofill completely encrypts the autofill process for native UI components in apps, and local storage with master password encryption is considered the best approach for web extensions.

While many password managers offer unique features and advantages, they also come with limitations. LastPass is plagued with autofill issues, outdated apps, limited free version features, and a 2022 data breach that tarnished its reputation. Dashlane is limited in its password storage options, simultaneous device usage, and cloud storage. LogmeOnce, a great freemium option, is overloaded with features and has a cluttered user interface, making it overwhelming for common users. BitWarden, an open-source solution, lacks TOTP and password sharing features in its free version. KeePassXC, another open-source option, lacks password sharing and can be difficult to configure manually. 1Password, NordPass, and Keeper are leaders in the password manager space but lack free versions, making them inaccessible to those unwilling to pay for their services. RoboForm, a popular and relatively inexpensive option, lacks TOTP support for its mobile apps and cloud sync options.

V. Addressing the Challenges with An Innovative Solution

Our software solution aims to provide a comprehensive password management system with a graphical user interface, developed using Android Studio. This system enables users to securely store their online credentials, including usernames and passwords, along with other sensitive information, in an encrypted database. The encryption is managed using advanced algorithms such as Rijndael

AES, Two-Fish, and ChaCha20. To access the system, users must login and the validity of the user is verified. Users can create and add to the database by using provided templates, and the list of accounts is organized into groups for easy access. Passwords are hidden by default but can be revealed upon selection. The software also includes a robust key-derivation function using Argon2/AES-KDF, making password handling secure.

In addition to storing passwords, our software also includes features such as zero-knowledge encryption, unlimited vault storage and synchronization, an open-source codebase, a secure password and passphrase generator, 2-factor authentication login, TOTP and HOTP support for stored credentials, and the ability to store notes, credit cards, and identities. The goal is to provide these advanced security features at a low cost to ensure that more people can benefit from proper management of their online credentials without compromising on security.

The system will be developed using Android Studio and programming languages such as Kotlin and Java, which are freely available online. The deployment platform will consist of Android devices, which are widely available, and laptops that meet the necessary requirements. The cloud server storage will be compatible with various solutions such as Google Drive, OneDrive, and Dropbox, as long as the corresponding app is installed on the user's smartphone. The only cost involved will be for internet access for downloading the required tools and for cloud storage synchronization.

Overall, this project is cost-effective as it utilizes freely available software and programming languages and only requires internet access for proper functionality. The goal is to keep the cost as low as possible to make the benefits of secure password management accessible to a wider audience.

VI. Design Implications for Solving Problems

Zero-knowledge encryption is a method of encryption where the encrypted data is stored on the server, and the decryption key is stored locally, on the user's device. This way, the server or any third party cannot access the decryption key and therefore cannot access the original data. In other words, the server has no knowledge of the original data, hence the name "zero-knowledge." This architecture provides enhanced security for sensitive data, as the encrypted data cannot be decrypted by anyone other than the user who has the decryption key.

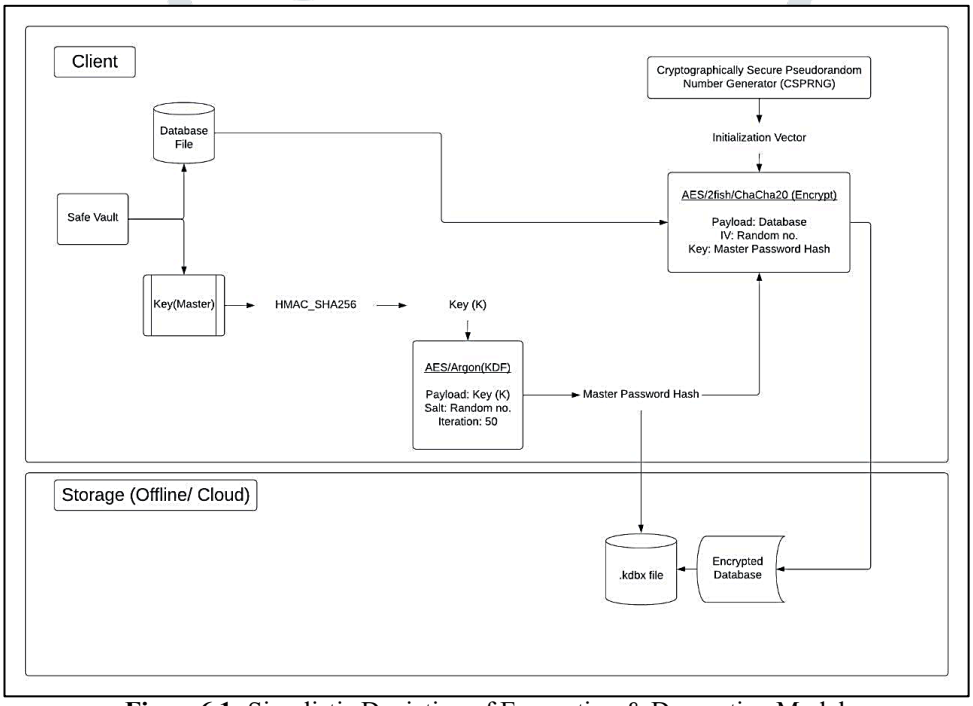

Figure6.1: Simplistic Depiction of Encryption & Decryption Model

Our system implements comprehensive encryption, which encompasses all elements of the database, including passwords, usernames, URLs, notes, and other relevant information. The encryption technique adopted leverages AES (Rijndael), Two-Fish, and ChaCha20 - widely recognized as secure encryption algorithms. The encryption is executed using the Cipher Block Chaining (CBC) mode, which conceals any patterns in the plaintext. Whenever the database is saved, a randomly generated Initialization Vector (IV) ensures the security of multiple databases encrypted with the same master key. The Encrypt-then-MAC approach confirms the authenticity and integrity of the data by creating a HMAC-SHA-256 hash of the ciphertext. This hash confirms the data's authenticity and integrity. To produce random bits for the high-level generation methods, we utilize a cryptographically secure pseudo-random number generator that is initialized using an entropy pool composed of various sources, such as the system cryptographic provider, current date/time, cursor position, operating system version, and more. The encryption master key is comprised of a combination of the master password, a key file, and/or a hardware key. The components of the master key are compressed using SHA-256, resulting in a 256-bit key (K). This key is transformed using a key derivation function with a random salt, making it difficult for attackers to carry out dictionary and guessing attacks. To further enhance security, the key derivation function (with a random salt) is utilized to prevent precomputation of keys and make dictionary and guessing attacks more challenging. This helps to ensure the security of the encryption process.

Securing a database involves making it difficult for unauthorized users to access its contents. This is accomplished through the transformation of the user's master key into a secure key using a key derivation function that includes a random salt. The more complex the key derivation function, the more challenging it will be for an attacker to guess the key. Key hashing transforms passwords into hash values, while key derivation converts passwords into keys used for encryption and decryption. These processes defend against attempts to steal passwords by intercepting the hash or key. AES-KDF and Argon2 are examples of supported key derivation functions. The AES-KDF iterates AES and can be adjusted by the user for increased difficulty. Argon2, the winner of the Password Hashing Competition, provides better protection against GPU/ASIC attacks, with variants Argon2d, Argon2id, and Argon2i. Argon2id is recommended for server applications as it provides better protection against side-channel attacks, while Argon2d is recommended for client devices as it offers better protection against GPU/ASIC attacks. The time required for key transformation may differ on different devices, so it's essential to ensure that all devices can load the database quickly.

Aside from dictionary attacks, password managers can also be targeted by keyloggers and process memory breaches. Process memory breaches focus on attacking the memory of an application where temporary data is kept. If a password manager is compromised in this way, an attacker can access sensitive information like passwords. This can happen if the password manager is running on a device with a security flaw or if there is a vulnerability in the password manager itself. For security and efficiency, it's crucial that sensitive data like entry passwords and master keys be encrypted when stored in memory. To heighten security, the password manager will close the database and only retain the file path and certain view parameters when the workspace is locked.

Password managers typically have an auto-fill feature that sends fake keystrokes to other applications, making it challenging for the target app to differentiate between real and simulated keypresses. However, this feature can be vulnerable to keyloggers as well. Keyloggers are malicious software or hardware that track every keystroke made on a computer. If a user's password manager is breached by a keylogger, the attacker can obtain their login information and potentially access sensitive information. To address this issue, the password manager uses the device clipboard to transfer parts of the auto-typed text to the target app, making it challenging for keyloggers to monitor the process. While this adds another layer of security, it's not foolproof and can still be vulnerable to spyware specifically designed to log obfuscated auto-types. It's important to keep in mind that there is no perfect security solution and the best way to protect sensitive data is to regularly update the password manager software and follow good security practices.

Database synchronization refers to the process of keeping multiple copies of a database in sync. This is an important feature in password managers, as it allows the user to access their passwords from any device. The password manager synchronizes the encrypted database across devices, ensuring that the latest version of the database is always available on all devices. This feature provides convenience for users who need access to their passwords on multiple devices, as well as increased security, as the encrypted database is stored in the cloud and can be retrieved in case the user loses their device.

The synchronization process of a password manager is accomplished through two main stages. Firstly, the data recovery from a shared source, which is overseen by a specific application. This application is responsible for transmitting and receiving files and making them available through a URI. The password manager itself acts merely as an editor, and any issues related to synchronization may stem from the cloud-based application used, such as difficulties with file conflicts or caching issues. The second stage involves the merging of updated information. Once new data is acquired through the URI, the password manager application is capable of combining it with the existing data. This allows the password manager to stay current and up-to-date, ensuring that users have the most accurate information at their disposal.

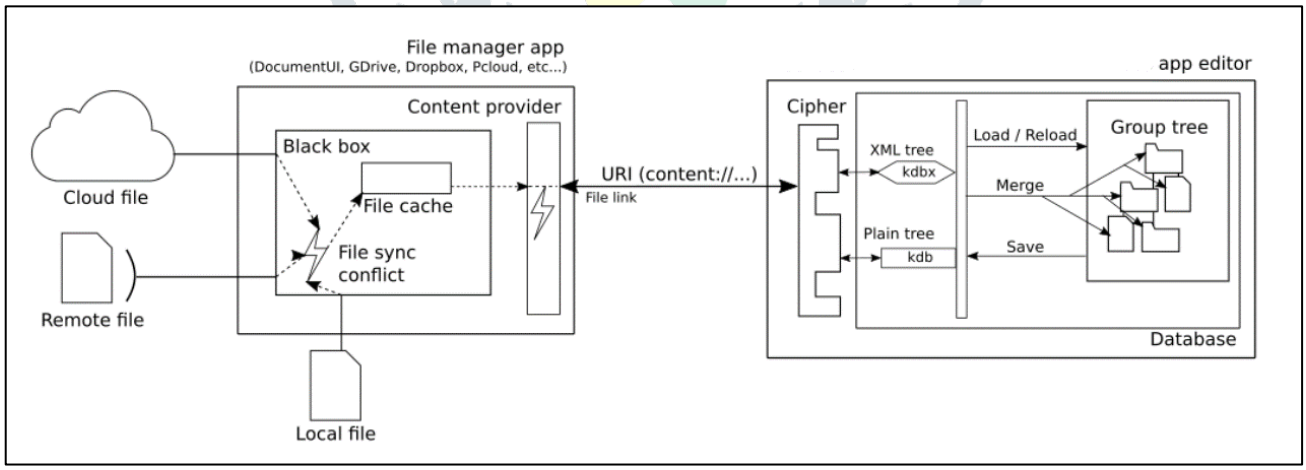

Figure6.2: Detailed Depiction of Database Synchronization Model

The password manager can provide several options for storing and synchronizing a user's password database. The database file can be stored locally on a device, on an external storage device, or on a remote file manager, such as a cloud application. The password manager acts as an editor for the database, but it is not responsible for managing data recovery or synchronization. This is typically handled by the cloud application used. The password manager can merge the retrieved data with the currently open data when a new version of the database is provided. This allows for greater scalability and configuration, but it also means that the user should be careful to choose a reliable cloud application for file management.

The main focus of the synchronization process is to decide which copy of an object is the latest one, mainly using the last modification time of the object. The synchronization process is performed at the entry level, ensuring that the combination of username and password is always consistent. In case of parallel updates and collisions, the password manager tries to store all information in an appropriate place. For example, if two users make changes to the same entry on two different devices and then try

to synchronize, the password manager will consider the entry on the device with the latest modification time as the current version, while storing the changes made on the other device as a history entry. This helps to prevent loss of data, while ensuring that the user always has access to the latest information.

In order to ensure seamless data synchronization, the password manager would provide users with several options for triggering a synchronization process. Manual synchronization is the most straightforward and simple way to keep data in sync. With manual synchronization, the user must actively initiate the process every time they wish to synchronize their data. This might be a good option for users who do not require real-time data synchronization or who are concerned about data privacy and security. Another way to trigger synchronization is through the use of the "Save" command. With this option, a user can save the changes they have made to their password manager database to a remote server or cloud storage. This provides a convenient way to ensure that all data changes are automatically backed up and synced with other devices. Triggers are another way to automatically initiate synchronization. A trigger is an event that occurs within the password manager that automatically initiates a synchronization process. For example, a trigger might be set to occur every time a password is added or updated, or every time the password manager is closed or reopened. This provides an easy and efficient way to keep data in sync without requiring manual intervention. Finally, scripting can also be used to initiate synchronization. Scripting allows users to automate various tasks within the password manager, including synchronization. This is particularly useful for advanced users who require more complex data synchronization processes, or for those who wish to automate the synchronization process as part of a larger workflow.

VII. Conclusion

Password managers serve a vital purpose in the safekeeping of our sensitive information that we manage online. They are software programs that allow individuals to create, store, and manage passwords securely. With the need to remember numerous unique passwords for various accounts and services, it can be challenging to ensure the security of this information. The occurrence of data breaches emphasizes the significance of having robust and original passwords that are difficult to crack.

There are various password managers available to users with varying features and security measures. Utilizing a password manager offers secure password storage, auto-generated passwords, password sharing options, and multi-factor authentication. The security of password managers also depends on the encryption methods, database storage techniques, and security protocols they employ.

To improve on the existing password managers, it is necessary to focus on strengthening the encryption mechanisms and database storing mechanisms. The encryption mechanism should use state-of-the-art encryption algorithms supporting post-quantum cryptography and use encryption keys that are stored locally on the device. The database storing mechanism should also be designed in a way that minimizes the risk of unauthorized access to sensitive information. Additionally, multi-factor authentication should be implemented to provide an additional layer of security. Furthermore, it is important to ensure that security methodologies such as IP whitelisting, bug bounty programs and regular independent audits are in place to minimize the risk of security breaches.

In essence, password managers play a vital role in securing sensitive information and protecting against data breaches. Improving the existing password managers would require a focus on encryption mechanisms, database storing mechanisms, and security methodologies, to provide the best possible security to users. With the right focus and resources, it is possible to create a secure and user-friendly password manager that provides peace of mind and protects against the growing threat of cyber-attacks.

VIII. Acknowledgement

We would like to express our gratitude to Jain (Deemed-to-be) University, Bangalore for providing us with the resources and support necessary to complete this review paper. Their contribution has been invaluable, and we are grateful for the opportunities and experiences that have been provided to us.

References

[1] E. A. Gallagher, "Choosing the Right Password Manager," *https://doi.org/10.1080/00987913.2019.1611310*, vol. 45, no. 1–2, pp. 84–87, Apr. 2019, doi: 10.1080/00987913.2019.1611310.

[2] S. K. Shinde and M. v Deshpande, "A Study for an Ideal Password Management System," vol. 10, 2022, doi: 10.22214/ijraset.2022.39970.

[3] D. Silver, S. Jana, D. Boneh, E. Chen, and C. Jackson, "Password Managers: Attacks and Defenses", Accessed: Feb. 05, 2023. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/silver

[4] A. Karole, N. Saxena, and N. Christin, "A comparative usability evaluation of traditional password managers," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6829 LNCS, pp. 233–251, 2011, doi: 10.1007/978-3-642-24209-0_16/COVER.

[5] "Password Security Design 1Password Memberships," Oct. 2021. Accessed: Feb. 05, 2023. [Online]. Available: https://1passwordstatic.com/files/security/1password-white-paper.pdf

[6] "1Password Watchtower." https://watchtower.1password.com/ (accessed Feb. 05, 2023).

[7] "Security audits of 1Password." https://support.1password.com/security-assessments/ (accessed Feb. 05, 2023).

[8] "CVE-2021-26905 for all versions of the 1Password SCIM bridge released prior to February 8, 2021." https://support.1password.com/kb/202102/ (accessed Feb. 05, 2023).

[9] "CVE-2020-10256 for all beta versions of the 1Password command-line tool and SCIM bridge released prior to December 24, 2018." https://support.1password.com/kb/202010/ (accessed Feb. 05, 2023).

[10] 8bit Solutions LLC, "Bitwarden Security Assessment Report ISSUE SUMMARIES, IMPACT ANALYSIS, AND RESOLUTION 8BIT SOLUTIONS LLC," Nov. 2018. Accessed: Feb. 05, 2023. [Online]. Available: https://cdn.bitwarden.net/misc/Bitwarden%20Security%20Assessment%20Report.pdf

[11] "Have you been pwned? | Bitwarden Blog." https://bitwarden.com/blog/have-you-been-pwned/ (accessed Feb. 05, 2023).

[12] "Compliance, Audits, and Certifications | Bitwarden Help Center." https://bitwarden.com/help/is-bitwarden-audited/ (accessed Feb. 05, 2023).
[13] I. BitWarden, "Bitwarden Network Security Assessment Report ISSUE SUMMARIES, IMPACT ANALYSIS, AND RESOLUTION BITWARDEN, INC," May 2022. Accessed: Feb. 05, 2023. [Online]. Available: https://bitwarden.com/images/resources/2022-bitwarden-network-security-assessment-report.pdf
[14] "Security: Bitwarden Desktop app grants RCE to Bitwarden developers. · Issue #552 · bitwarden/desktop." https://github.com/bitwarden/desktop/issues/552 (accessed Feb. 05, 2023).
[15] "XChaCha20 encryption | NordPass." https://nordpass.com/features/xchacha20-encryption/ (accessed Feb. 05, 2023).
[16] nordvpn S.A., "NordPass Business Whitepaper | NordPass," May 2020. Accessed: Feb. 05, 2023. [Online]. Available: https://nordpass.com/nordpass-business-whitepaper.pdf
[17] "Data breach scanner: check your password safety | NordPass." https://nordpass.com/features/password-breach-report/ (accessed Feb. 05, 2023).
[18] "NordPass completes a comprehensive security audit | NordPass." https://nordpass.com/blog/nordpass-security-audit-2020/ (accessed Feb. 05, 2023).
[19] "Nord VPN-6.31.13.0 - 'nordvpn-service' Unquoted Service Path - Windows local Exploit." https://www.exploit-db.com/exploits/48790 (accessed Feb. 05, 2023).
[20] "The ultimate security for your passwords | Zoho Vault." https://www.zoho.com/vault/security.html (accessed Feb. 05, 2023).
[21] Zoho Corporation Private Limited, "Zoho Vault Security Specifications ." Accessed: Feb. 05, 2023. [Online]. Available: https://www.zoho.com/sites/default/files/zoho-vault-security-specifications.pdf
[22] "Complete Insights With Our Visual Reports | Zoho Vault." https://www.zoho.com/vault/vault-security.html (accessed Feb. 05, 2023).
[23] "IP Restriction | Zoho Vault." https://help.zoho.com/portal/en/kb/vault/admin-guide/articles/vault-configure-enable-ip-restriction (accessed Feb. 05, 2023).
[24] "LastPass Technical Whitepaper." Accessed: Feb. 05, 2023. [Online]. Available: https://assets.cdngetgo.com/da/ce/d211c1074dea84e06cad6f2c8b8e/lastpass-technical-whitepaper.pdf
[25] "How do I set up biometrics and mobile account recovery on Android for LastPass? - LastPass Support." https://support.lastpass.com/help/how-do-i-set-up-and-use-mobile-account-recovery-on-android-lp010120 (accessed Feb. 05, 2023).
[26] "Dark Web Monitoring & Alerts | LastPass." https://www.lastpass.com/features/dark-web-monitoring (accessed Feb. 05, 2023).
[27] P. Berba, "How to decrypt a LastPass vault. This is a medium-sized extract from a… | by Pepe Berba | Medium," May 31, 2016. Accessed: Feb. 05, 2023. [Online]. Available: https://medium.com/@pberba/how-lastpass-decrypts-your-vault-279153350930
[28] "CVE - Search Results." https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=LastPass (accessed Feb. 05, 2023).
[29] D. Reichl, "Security - KeePass," Jul. 2015, doi: 10.6028/NIST.FIPS.180-4.
[30] "KeePass Gets Full Biometrics Support in the Latest Keepass2Android Update « Android :: Gadget Hacks." https://android.gadgethacks.com/how-to/keepass-gets-full-biometrics-support-latest-keepass2android-update-0331354/ (accessed Feb. 05, 2023).
[31] "EU to fund bug bounty programs for 14 open source projects starting January 2019 | ZDNET." https://www.zdnet.com/article/eu-to-fund-bug-bounty-programs-for-14-open-source-projects-starting-january-2019/ (accessed Feb. 05, 2023).
[32] "CVE - Search Results." https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=keepass (accessed Feb. 05, 2023).
[33] "Keeper Encryption Model - Enterprise Guide." https://docs.keeper.io/enterprise-guide/keeper-encryption-model (accessed Feb. 05, 2023).
[34] "Login to Keeper on macOS with Touch ID - User Guides." https://docs.keeper.io/user-guides/tips-and-tricks/login-to-keeper-on-macos-with-touch-id (accessed Feb. 05, 2023).
[35] "Dark Web Monitoring for Business - Keeper Security." https://www.keepersecurity.com/breachwatch.html (accessed Feb. 05, 2023).
[36] "Data Security: Safeguard Sensitive Data with Enpass." https://www.enpass.io/security/ (accessed Feb. 05, 2023).
[37] "Quick Unlock — Enpass Security Whitepaper documentation." https://support.enpass.io/docs/security-whitepaper-enpass/quick_unlock.html (accessed Feb. 05, 2023).
[38] G. Leo, S. Consultant, and F. Watson, "Enpass Apps-Security Assessment," Nov. 2018. Accessed: Feb. 05, 2023. [Online]. Available: https://dl.enpass.io/docs/EnpassSecurityAssessmentReport.pdf
[39] "Vulnerability-Reporting - Enpass." https://www.enpass.io/vulnerability-reporting/ (accessed Feb. 05, 2023).
[40] "Keychain data protection - Apple Support (CA)." https://support.apple.com/en-ca/guide/security/secb0694df1a/web (accessed Feb. 05, 2023).
[41] "Secure iCloud Keychain recovery - Apple Support (CA)." https://support.apple.com/en-ca/guide/security/secdeb202947/web (accessed Feb. 05, 2023).
[42] "Accessing Keychain Items with Face ID or Touch ID | Apple Developer Documentation." https://developer.apple.com/documentation/localauthentication/accessing_keychain_items_with_face_id_or_touch_id (accessed Feb. 05, 2023).
[43] "CVE - Search Results." https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Apple+Keychain (accessed Feb. 05, 2023).
[44] I. Dashlane, "Dashlane's Security Principles & Architecture," Aug. 2022. Accessed: Feb. 05, 2023. [Online]. Available: https://www.dashlane.com/download/whitepaper-en.pdf

[45] "Security alerts and Dark Web Monitoring in Dashlane – Dashlane." 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 (accessed Feb. 05, 2023).
[46] P. Gentili, S. Shader, R. Yip, and B. Zeng, "Security Analysis of Dashlane," 2016, Accessed: Feb. 05, 2023. [Online]. Available: https://courses.csail.mit.edu/6.857/2016/files/25.pdf
[47] "Dashlane - DLL Hijacking - Windows local Exploit." https://www.exploit-db.com/exploits/44066 (accessed Feb. 05, 2023).
[48] "Why Use logmeonce - LogMeOnce." https://www.logmeonce.com/why-use-logmeonce/ (accessed Feb. 05, 2023).
[49] "Vulnerability Disclosure Policy - LogMeOnce." https://www.logmeonce.com/vulnerability-disclosure-policy/ (accessed Feb. 05, 2023).
[50] "Password Shock - LogMeOnce." https://www.logmeonce.com/password-shock/ (accessed Feb. 05, 2023).
[51] "RoboForm Security Overview Whitepaper," Oct. 2019. Accessed: Feb. 05, 2023. [Online]. Available: https://www.roboform.com/pdf/RoboForm_Security_White_Paper.pdf
[52] "Key Features." https://www.roboform.com/key-features (accessed Feb. 05, 2023).
[53] "Zero-Day Research | Fixes Pending | FortiGuard." https://www.fortiguard.com/zeroday?type=zd&vendor=Roboform (accessed Feb. 05, 2023).
[54] "RoboForm Manual Android." https://www.roboform.com/manual-android (accessed Feb. 05, 2023).
[55] "DualDAR Encryption | Knox Platform for Enterprise White Paper." https://docs.samsungknox.com/admin/whitepaper/kpe/DualDAR.htm (accessed Feb. 05, 2023).
[56] "General questions and information about Samsung Pass." https://www.samsung.com/us/support/answer/ANS00066601/ (accessed Feb. 05, 2023).
[57] "ZDI-19-515 | Zero Day Initiative." https://www.zerodayinitiative.com/advisories/ZDI-19-515/ (accessed Feb. 05, 2023).
[58] I. Keeper Security, "Keeper MSP Technical Whitepaper," 2019. Accessed: Feb. 05, 2023. [Online]. Available: https://www.keepersecurity.com/assets/pdf/Keeper-Managed-Service-Provider-Tech-WhitePaper.pdf