I. J. Mathematical Sciences and Computing, **2024, 4, 21-33**
Published Online on December 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2024.04.03

An Automated Solution for Secure Password Management

Prageeth Fernando*
Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura, Sri Lanka
E-mail: prageethfndo@gmail.com
ORCID ID: https://orcid.org/0000-0001-7567-6863
*Corresponding Author

Navinda Dissanayake
Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura, Sri Lanka
E-mail: navinda@yandex.com
ORCID ID: https://orcid.org/0009-0000-6903-7846

Venusha Dushmantha
Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura, Sri Lanka
E-mail: venusha125@gmail.com
ORCID ID: https://orcid.org/0009-0004-1457-9778

Chamara Liyanage
Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura, Sri Lanka
E-mail: dlchamara@gmail.com
ORCID ID: https://orcid.org/0000-0003-3833-3911

Chamila Karunatilake
Department of Information and Communication Technology, Faculty of Technology, University of Sri Jayewardenepura, Sri Lanka
E-mail: itchamila@gmail.com
ORCID ID: https://orcid.org/0000-0002-7733-2324

Received: 04 March, 2024; Revised: 02 August, 2024; Accepted: 28 September, 2024; Published: 08 December, 2024

Abstract: This research paper introduces *SecretCentric*, an innovative automated hardware-based password management system addressing the challenges of widely used password authentication methods, which have long been criticized for their poor performance. Password management plays a crucial role in protecting users' digital security and privacy, with key factors including password generation, storage, renewal, and reuse mitigation. Although numerous password managers and solutions have been introduced to tackle these challenges, password management automation has never been thoroughly explored. This study aims to revolutionize the field by eliminating the burden of manual password management from users by automating the entire process. Upon concluding a comprehensive survey, insights into user perceptions of password management and prevalent malpractices were identified. *SecretCentric* was designed to maximize the security and usability trade-off aligning with identified user expectations. Preliminary evaluations indicate that *SecretCentric* offers significant improvements over existing options, highlighting the necessity for an automated solution that balances security and usability in the era of increasing online services. The system's success demonstrates the importance of proper password management rather than replacement, contributing to research advancement in user authentication and credential management.

Index Terms: Password, Password manager, Authentication, Automation.

1. Introduction

Passwords are still being used as the primary authentication mechanism for information systems since coming into existence over sixty years ago although it is the oldest mechanism which criticized for poor security performance [1]. Over 30% of security breach incidents were caused by malpractice related to passwords and they often result in financial losses, leakage of personal data, and unauthorized access to private systems. Among reported statistics, 45.7% of users practice password reuse and 62.9% of users only renew their passwords when they are prompted to do so [2]. Though there have been many attempts to find alternatives for passwords such as biometrics, patterns, security tokens, PIN, OTP, and SSO, none of them managed to completely replace passwords as a result of their weaknesses and other circumstantial factors. As a result, 2FA and MFA were introduced to increase the security performance of password-based authentication. But still, passwords are the first factor of Two-Factor and Multi-Factor authentication schemes.

The exponential growth of the Internet and the growing number of web applications have led users to maintain 12 to 26 password-protected accounts [3,4]. Cybersecurity experts suggest that strong password policies and rules should be applied to force users to create complex and non-guessable passwords to secure computer systems and networks. Users now have the challenge of remembering multiple complex passwords for different accounts when the human being is generally poor at memorizing more than seven characters or digits [5,6]. Therefore, password management has become a complex task for an average computer user and users tend to reuse the same set of passwords with or without small modifications to the existing ones while forgetting about periodic password renewal. With the purpose of addressing this ever-lasting password management problem, password managers were introduced to generate strong passwords and securely store them. These password managers can be divided into software-based and hardware-based managers. Most of the existing password management solutions either rely on cloud infrastructure which is often viewed as a target for bad actors or lack usability and features to be useful for an average user. Additionally, there is a significant number of users who do not trust cloud services enough to store their sensitive data. Due to the nature of the existing solutions, they have failed to provide a proper solution to this password management problem [5]. Though there are many studies and attempts to solve this problem, none of them have managed to come up with a solution that is secure, usable, completely offline, and automated. A solution where the end-user does not have to worry about managing passwords through some complicated process, but rather "just configure it once and then forget about it".

This paper is focused on proposing an automated solution for password management named *SecretCentric*, which mitigates password reuse and encourages periodic password renewal while providing other password management processes such as complex password generation, secure storage, etc. This solution consists of a completely offline hardware wallet (SC-wallet), client-side software (SC-client), and a web browser extension (SC-extension). This solution is motivated by providing automated password management for users while removing the burden of manual password management from users. This also provides a better solution for users who are reluctant to move to password managers due to the safety concerns of their sensitive data. To fill the gaps between password management problems, this proposed solution is completely immune to cloud-based security issues, and passwords are stored and transferred in encrypted form to increase security. Users can authenticate the device simply by using their fingerprint and the device will take care of creating, renewing, auto-filling, and storing passwords for all web applications that require login functionality.

In an era where we cannot avoid using web services and platforms that require password-based authentication, this proposed system addresses the issues with existing password management solutions while providing a better mechanism for managing passwords more conveniently and securely. Users can set it up once and use it without a second thought due to built-in automation capabilities. This research provides great conceptual, as well as theoretical significance by not viewing passwords as something that must be replaced, but as something that needs to be managed and secured properly. This also diverges this study from similar studies in the field by addressing a common problem using existing technologies rather than trying to reinvent the wheel. Furthermore, the practicality of this research is that instead of putting faith in the user, it focuses on automating the hardware device to perform everything for them. This makes it very simple for an average user to start using this device. Moreover, this study focuses on using encryption technologies and secure communication protocols to manage everything offline while providing easy access to the data. The standard encryption algorithms such as AES-256 in conjunction with custom algorithms to secure the hardware wallet and provide secure communication between user devices will contribute to the advancement of research in this domain and shape new avenues to find better solutions to solve security concerns regarding user authentication and credential management.

2. Background

2.1. Password problem

Taneski, Heričko, and Brumen [1] identified three user authentication methods in modern computer systems: "what user knows" (textual and graphical passwords), "what user has" (smart cards, tokens), and "what user is" (biometrics). Despite being just one aspect of system security, passwords are crucial [7]. Yıldırım and Mackie [8], along with Habib et al. [9], asserted that textual passwords are cost-effective, easy, and quick, with no compatibility or technical issues.

Consequently, they are widely used as the primary authentication mechanism. Lyastani et al. [10] also found textual passwords to be the de facto authentication scheme online. Although graphical passwords offer better security, they take longer to register and log in [1]. Yıldırım and Mackie [8] noted that while many alternatives have been researched, none have surpassed the simplicity of textual passwords, which remained popular 61 years after the Compatible Time-Sharing System (CTTS) in 1961 [11].

Over the past decade, the rapid growth of the internet and web applications has led to an increase in password-protected user accounts [12]. On average, internet users maintain 12 to 26 accounts, while undergraduates maintain an additional 8 academic accounts, often forgetting new passwords within 12 hours [3,4]. Effective password management, which involves strong creation, mitigation of reuse, memorization, renewal, and availability, relies heavily on memorization [13]. Gao et al. [3] discussed psychological theories relevant to password memorization, highlighting how human memory limitations often result in pattern-based passwords. Glory et al. [14] noted that manually created or randomly generated passwords can be insecure or easily forgettable. They proposed an automated system that generates robust, user-friendly passwords based on user-provided information (e.g., favorite novels, and secret dates). Tests showed these passwords met minimum strength criteria and resisted various attacks, though the system failed to defend against social attacks.

Taneski, Heričko, and Brumen [1] suggested that users create easily guessable passwords due to the difficulty of memorizing high-entropy ones. Woods and Siponen [15] found that repeating password verification can increase memorability from 42% to 70%, though it may be inconvenient. Fredericks [11] presented that manually created passwords, even when adhering to password policies, can be vulnerable to various attacks. Singh and Raj [16] argued that password strength checkers, while helpful, suffer from inconsistency, accuracy issues, and biases due to static policies. These checkers also cannot measure usability or memorability. They proposed an algorithm that dynamically generates password policies based on character frequency and uses special characters to increase complexity, making it harder to crack passwords with the dictionary and rainbow attacks. Biesner et al. [17] proposed a deep learning technique for generating novel, realistic passwords to outperform existing password generation algorithms. Smith [18] highlighted that people's bias towards simple passwords with a word and a number significantly reduces security and leaves accounts vulnerable to dictionary attacks. Three random password generation schemes (ALPHANUM, DICEWARE, and PRONOUNCE3) can enhance entropy and security. Grilo, Ferreira, and Almeida [19] studied 15 password managers, focusing on Google, Bitwarden, and KeePass algorithms since they are open-source and widely used. Generated passwords must satisfy composition policies to prevent easy guessing or reuse. Google and Bitwarden algorithms support string permutation, while KeePass generates random characters from defined sets. To ensure security, generated passwords should have a uniform distribution within the same policy.

Gao et al. [3] showed that average users maintain 12-26 online accounts but use only 7 unique passwords, leading to password reuse. Kuka and Bahiti [20] and Rahalkar and Gujar [12] found that 52%-56% of people overestimate their manually created passwords and reuse them with little modification. Stobert and Biddle [21] reported that 75% of respondents admitted to reusing passwords for convenience and speed. Tatli and Seker [22] identified 43 password reuse patterns from a 14.5 million password dataset. The conflict between IT professionals' strict password policies and users' desire for convenience results in password reuse [5]. Habib et al. [9] revealed that only 20% of services require periodic password updates, and 67% of users create new passwords by modifying existing ones. Abuzaraida and Zeki [23] found that only 8% of users regularly renew passwords, and just 6% do so twice a year. Lyastani et al. [10] attributed password reuse across multiple accounts to the limited capacity for remembering multiple passwords.

2.2. Password alternatives

Morris and Thompson [7] identified textual passwords as weak and a threat to security, prompting the search for alternatives. Chaudhary et al. [5] listed various types of password practices, and numerous alternative solutions have been implemented, such as biometrics, smart cards, and OTP. However, none of them have replaced passwords due to their limitations and vulnerabilities. Siddique, Akhtar, and Kim [24] suggested that combining passwords with other approaches would improve security, with 84% of survey participants supporting the elimination of passwords and 76% preferring alternative authentication. Zimmermann and Gerber [25] found 85 authentication approaches and categorized them, concluding that password and biometric combinations outperform others in security, usability, and deployability. Kurniawan et al. [26] proposed a methodology for user authentication using OTP with AES and Blowfish algorithms, which can increase the security of username-password authentication but not completely replace it.

2.3. Use of password managers

Users face the challenge of remembering multiple complex passwords, with humans generally struggling to memorize more than seven characters or digits. Strict password policies are often unrealistic and time-consuming, causing cognitive overload [5,27]. To address this issue, password managers have been developed, which store and generate strong, unique passwords while requiring users to remember only a single master password [6,10,12,19,28]. However, password managers can be vulnerable to various attacks and suffer from a single point of failure due to the master password. There are different types of password managers, including retrieval and generative password managers, desktop, online, mobile-based, and USB-based [29,30]. While software-based password managers are more cost-effective and easier to deploy, hardware-based password managers are still being developed to overcome their

limitations [6].

Software-based password managers operate in two modes: online and offline. Offline managers, or desktop managers, store encrypted password vaults on client-side devices, requiring users to transport the device. Online managers store encrypted vaults on a cloud service or remote location, accessible via the internet. Despite their convenience, password managers have low adoption rates, with 60%-63% of users relying on memorization for password management and only 8% using password managers. This low adoption is due to developers' focus on specific features rather than usability and security measures [5,12]. Password managers should have strong security architecture and mechanisms to protect users' sensitive data. Offline managers face challenges such as device portability, single-point-of-failure, and potential unencrypted passwords in temporary files, which contribute to their low adoption rate among non-expert computer users [4,31].

Online password managers address portability, synchronization, and single-point-of-failure issues by storing encrypted password vaults in the cloud or remote locations, allowing users to access their credentials anytime and anywhere [32,33]. However, these managers can be vulnerable to security breaches, mistrust of service providers, and critical vulnerabilities in browser extensions [5,13,19]. Online managers can be open-source or closed-source. Open-source managers allow users to examine code, report vulnerabilities, and choose a preferred server but may expose vulnerabilities to attackers and require technical expertise for server configuration. Closed-source managers require users to trust the company behind the service, with the proprietor responsible for updates and security. Closed-source managers may attract more attackers due to their proprietary nature, as seen with LastPass in 2017 [13,34].

2.4. Hardware-Based Password Managers

There is a research gap between software-based and hardware-based password managers, with no existing hardware solution that is portable, cost-effective, and offers full control over stored credentials [6]. Hardware wallets should interact with user web browsers through various communication channels [35]. Stajano [36] proposed a hardware device called Pico, which transforms authentication from "something the user knows" to "something the user has." Pico communicates using public and private key encryption and supports continuous authentication and identification. It is theft-resistant and loss-resistant, utilizing a docking station for encrypted backups. Pico aims to provide usability and security without replacing passwords, as other mechanisms often trade-off between the two. However, a study on Pico's usability revealed participants disliked QR code scanning, preferring an alternative authentication method, but appreciated the automation of password management [35].

Gupta et al. [29] proposed a USB-based Portable Password Manager using an Arduino Micro microcontroller for secure storage and communication with user devices through USB. It also features an OLED display and authentication node for continuous user identification. Wang and Khan [34] suggested a hardware-based token authentication system that uses a tamper-resistant chip, USB and NFC dual interface, and QR codes. However, vulnerabilities include potential JavaScript file injections and a lack of a loss-resistant mechanism. Guo et al. [30] introduced PUFPass, a hardware-based password manager that uses Physical Unclonable Function (PUF) for enhanced security. It supports QR codes for transportability and device migration but has drawbacks, such as non-recoverable passwords and increased complexity in password management. PUF-based solutions also struggle with backup and restoration, portability, and reliance on cloud services, raising security concerns [37,38].

Naing Oo [6] introduced E2PM, a novel hardware password manager stored on a regular USB stick, which can be accessed through the Midori web browser. The system is comprised of two main components: the Core System, a runtime image, and the Secure Data Partition, a memory that allows reading and writing. To use E2PM, users must either live boot the USB drive or establish a connection through the VirtualBox Virtual Machine manager. E2PM satisfies Stajano's [36] requirements of memoryless, scalable, and secure authentication methods. However, it is not very user-friendly for non-technical users, as it is not hot-pluggable like other hardware wallet managers. Despite this drawback, E2PM offers increased deployability due to its simplicity in hardware and software requirements. In the event of wallet theft, attackers would face difficulty accessing the password vault, which is protected by a 16-character master password. Nevertheless, the system is vulnerable to Cold Boot Attacks, during which an attacker could potentially find the master key stored in the RAM. Unfortunately, E2PM does not provide any loss resistance, as there is no backup or migration mechanism available for recovering lost data.

3. Methodology

SecretCentric has three major components namely, SC-wallet, SC-client, and SC-extension as shown in Figure 1. SC-wallet is the hardware device that stores user passwords and other data, SC-client is a client-side application that communicates with SC-wallet, and SC-extension is a custom-developed web browser extension that allows communication between SC-client and web browser.

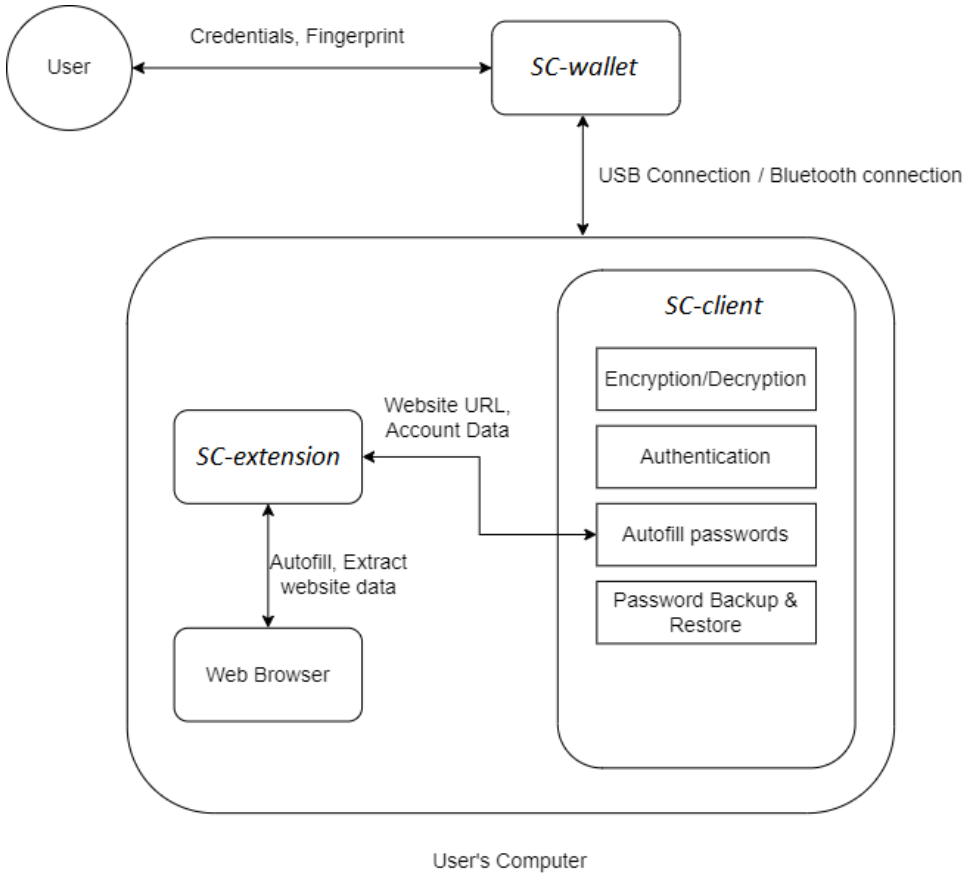

Fig. 1. Proposed solution overview

3.1. SC-wallet architecture

The SC-wallet consists of a Raspberry Pi microcontroller which is capable of handling database transactions, user authentication, and device connectivity. A fingerprint scanner is used to read the fingerprint to authenticate and validate the ownership. The fingerprint scanner is connected to the Raspberry Pi via a USB connection to transfer data between the scanner and the microcontroller. It uses the UART protocol for communication. SC-wallet always stays in locked mode until it gets an authentication request from the SC-client application program. There are two types of connectivity modes, Bluetooth as the wireless method and USB as the wired communication method. SC-wallet is a small portable device that consists of a 5v5000mAh li-ion rechargeable battery which can power up the device and providing Bluetooth connectivity. When SC-wallet is wired to a user device, both power and communication happen through the USB link.

3.2. SC-client architecture

SC-client is an application program that can be installed on user devices to function with the SC-wallet. It acts as an intermediary layer between the SC-wallet and the web browser. This handles all processes of unique password generation, renewal, reuse mitigation, backup, autofill, migration, and encryption/decryption.

3.3. Authentication & password autofill

Once the user gets registered, a symmetric key is generated inside of the SC-wallet, and it will be used for all encryption and decryption tasks inside of the wallet. This key will not be stored and will only be available until the credential request is handled. The encrypted login credentials are stored in an SQLite database. When a credential request is made by SC-client, a database SELECT query is initiated, and symmetric-key decryption is applied. Decrypted credentials then will be transmitted to the SC-client through a public-key encrypted tunnel between SC-wallet and SC-wallet. SC-client will decrypt the data with its private key and pass it to the SC-extension to perform autofill. SC-extension has an interface to access the information of the visited web application, and it extracts data such as URL, request information, and login fields.

3.4. Backup & restore

Physical devices can be lost or stolen. But *SecretCentric* is theft-resistant since all data are encrypted and inaccessible without the respective fingerprint. However, it’s necessary to have a backup feature to securely back up all

user credentials and restore them in case of theft or loss. For that process, this solution is intended to use a novel mechanism. This system allows backup only for one device bound to the MAC address of the registered computer. So, third parties cannot make backups without the original computer. There may be situations where the user wants to change their computer. Since the device is bound with the MAC address of the computer, the existing MAC should be removed from the SC-wallet. Then the new computer's MAC address should be registered with the SC-wallet once connected to the new computer. This feature is handled by the SC-client.

Once the backup request is made, the SC-client sends the computer MAC address to the SC-wallet for validation. If validated, SC-wallet sends the database SQL file to the SC-client through a public-key encrypted tunnel. Then SC-client decrypts the SQL file and symmetrically encrypts it again before storing it on the user's computer. This symmetric encryption key consists of two subkeys. One subkey is stored inside the SC-wallet and given to the user to keep it secure to use in case of theft or loss. The second subkey is stored with SC-client. The system creates the symmetric key by XOR computing these two subkeys. The resultant key is used by AES-256 to encrypt the backup SQL file. When the user wants to restore a backup, the credential database is decrypted using the resultant key and sent to the SC-wallet.

3.5. Unique password generation

The SC-client can generate unique passwords with a novel built-in function. It utilizes two UUIDs as the foundation layer of the algorithm to make it more randomized and four policies are defined. Each password contains between 12 and 16 characters consisting of lowercase and uppercase English alphabet letters, special characters, and numbers. These are defined as classes in this algorithm. A policy defines that every password must contain at least 3 characters from each class which is randomly decided by the algorithm depending on the length of the password and the timestamp. Numbers and lowercase characters are directly taken from the two UUIDs. There are two dictionaries for upper-case letters and special characters. The random characters of the UUIDs are used as the indices to select upper-case letters and special characters within the respective dictionary.

3.6. Approach

The main goal of this study is to completely automate the password management process while mitigating password reuse and enforcing periodic password renewal. The SC-wallet includes a Raspberry Pi 2 Model B, chosen for its cost-effectiveness, adequate performance for encryption and database operations, stability, and power efficiency, along with an R307 fingerprint sensor for fast and robust secure user authentication. The Raspberry Pi module is installed with Raspberry Pi OS and SQLite DBMS with other necessary dependencies for the fingerprint sensor connectivity and functionalities. The SC-wallet can be connected to a personal computer through a USB cable and can only be accessed through the SC-client. When the user registration happens for the first time to a virgin SC-wallet, the user information should be provided through the SC-client and the preferred fingerprint will be captured through the fingerprint sensor. Provided user information including the fingerprint is stored in the SQLite database and whenever required to connect to the SC-wallet, the SC-wallet must be connected to a computer and get authenticated by providing the fingerprint. All stored user credentials in the database are AES-256 encrypted through a custom-developed algorithm that uses a portion of the registered fingerprint as the encryption key. All data communication between SC-wallet and SC-client are public key encrypted and OpenSSL certificate renewal is also done automatically. Account credentials are stored as desired, and SC-client creates 16-character long strong passwords for each new account. When the user wants to log in to an account, SC-extension, which can be installed on the browser, autofill the username and password instantly. The communication between SC-client and SC-extension is also protected by public key encryption with a monthly auto-renewing OpenSSL certificate. In case of unavailability of the operational extension, users can securely get the passwords to the clipboard and paste them manually. All credentials stored in the SC-wallet are frequently checked for age expiration and potential exposures. When an unsecured login session is created or the password age has expired, users are directed to renew the passwords. The whole login process and renewal process (autofill the current password and new passwords) are done automatically by the collaboration of SC-client and SC-extension. In case the SC-wallet was lost or stolen, a virgin SC-wallet can be restored from a backup through the SC-client on the personal computer that was initially used to register the previous SC-wallet. Frequent database backups are taken as SQL files and saved only on the initial device. SC-client encrypts the backup files with the AES-256 algorithm and the MAC address of the user's personal computer is bound to the SC-wallet to prevent the scarcity of backups or potential security threats.

4. Results

In this study, a survey was conducted over two months, involving 165 participants from diverse demographics in terms of age, gender, and computer literacy. This sample size was sufficient to capture meaningful insights into password management behaviors and challenges, consistent with similar studies. The survey aimed to explore usability and security trade-offs in users' perceptions and habitual attitudes toward password management. This section presents the findings, including demographics, computer literacy, user perceptions, habitual attitudes, experiences with password managers, and significant trends or patterns in the data.

Table 1. Demographics of respondents

Gender	Number of responses	Percentage
Male	95	57.3
Female	70	42.7
Age	**Number of responses**	**Percentage**
10-19	11	6.7
20-29	86	52.1
30-39	41	24.8
40-49	22	13.3
50-59	5	3

Table 2. Computer literacy of respondents

Literacy level	Percentage
Fundamental - typing, mouse	2.4
Basic - office package, internet & email	31.5
Intermediate - OS/hardware installation and management	32.1
Advanced - graphics, DBMS, programming & scripting	25.5
Proficient – network/system management & troubleshooting	8.5

As shown in Table 1, the majority of respondents (57.3%) were male, while 42.7 percent were female. The 20-29 age group had the highest total of respondents (52.1%), followed by the 30-39 age group (24.8%). Table 2 indicates that the majority of the survey respondents are well-educated (69.1%) and have intermediate to proficient computer literacy (66.1%).

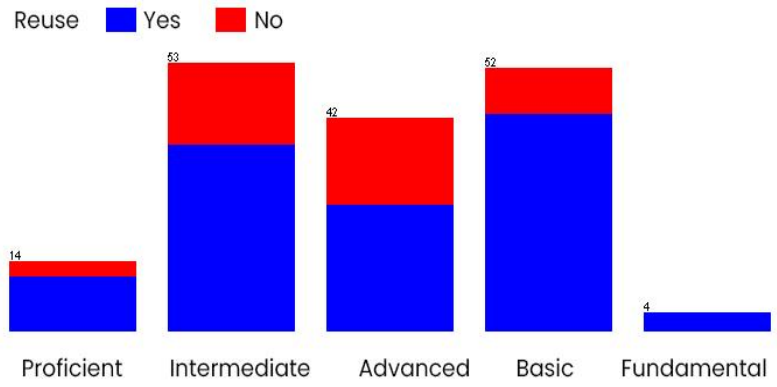

Fig. 2. Password reuse behavior based on computer literacy

Figure 2 presents how the survey respondents of different computer literacy levels practice password reuse. Among the total 165 respondents, 72.7% of respondents practice password reuse and most significantly, the majority (75.8%) of computer literate respondents (intermediate to advanced computer literacy levels) are also included within the respondents who practice password reuse.

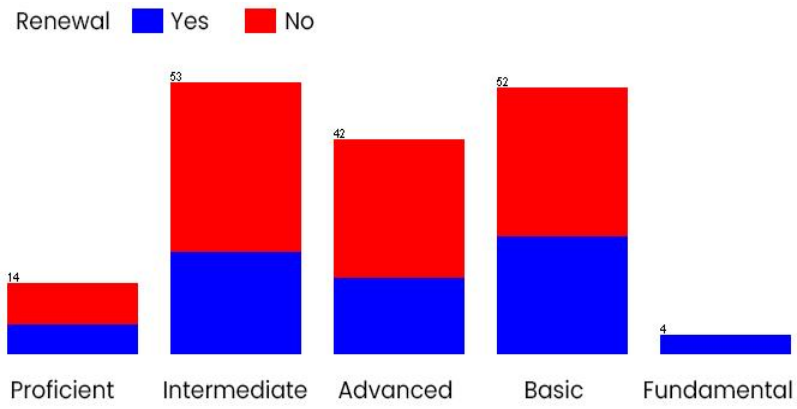

Fig. 3. Password renewal behavior based on computer literacy

Periodical password renewal practice among survey respondents in different computer literacy levels is shown in Figure 3. Among the total respondents, 41.8% of respondents practice periodical password renewal, and most significantly, the majority (69.5%) of computer-literate respondents are not included within the respondents who practice periodical password renewal.

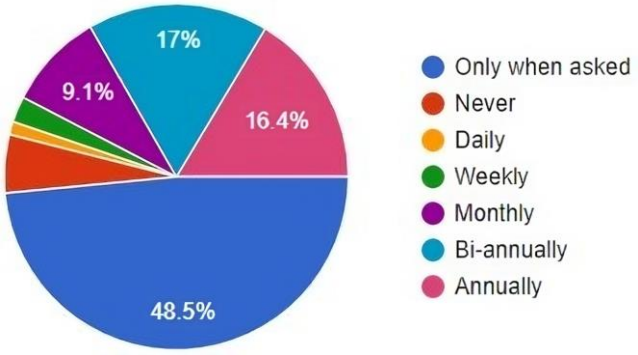

Fig. 4. Password renewal frequency

Figure 4 presents the frequency of survey respondents renewing their passwords and it clearly shows that the majority of respondents (48.5%) only change their passwords only when they are forced. 16.4% and 17% of respondents renew their password annually and bi-annually respectively. Only 9.1% of respondents mentioned renewing their passwords monthly.

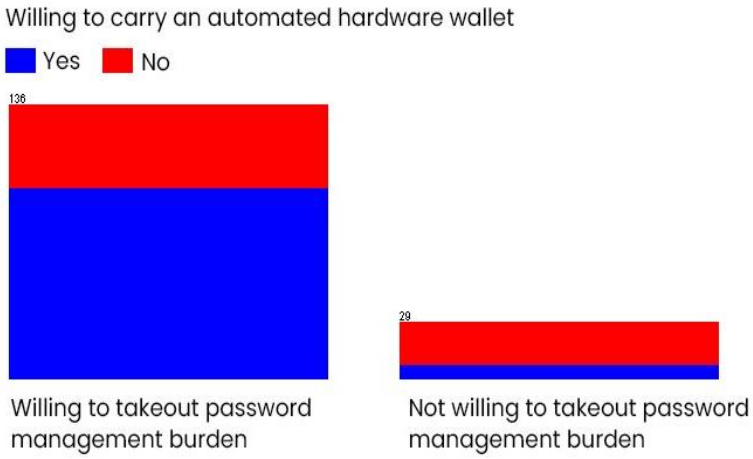

Fig. 5. Comparison of password management burden and automated hardware wallet adoption

Figure 5 presents the survey respondents' perception of password management and how many of them wish to have an automated solution for password management. Out of the 165 respondents, 85 respondents said that password management is very difficult for them, 67 respondents said that password management is easy for them and only 13 respondents said that they do not care about password management. Among the 85 respondents, who are facing the difficulties of managing passwords, 58 respondents are willing to carry a hardware wallet with the promise of taking out the password management burden from them. Furthermore, of the 67 respondents who said that password management is easy for them, 31 respondents are willing to face the burden of carrying a hardware wallet with them. 8 respondents are neither willing to care about password management nor to have an automated solution to take out the burden.

Table 3 shows the comprehensive time measurements of the proposed *SecretCentric's* major operations. Device authentication was dependent on the user's speed of providing the fingerprint and sensor performance. Password generation was tested on the https://signup.mail.com/ webpage, and both database request handling and autofill operations were tested on the https://www.facebook.com/login/ webpage.

Table 3. SecretCentric's operational time analysis

Operation	1st try	2nd try	3rd try	4th try	5th try	Average time
Device authentication	3s	2.49s	3.23s	2.53s	2.92s	2.834s
Password generation	0.641ms	0.558ms	0.595ms	0.570ms	0.543ms	0.581ms
Database request handling	2ms	4ms	3ms	2ms	2ms	2.6ms
Autofill	262ms	244ms	274ms	276ms	258ms	262.8ms
Backup	1.42s	1.67s	1.48s	1.52s	1.55s	1.57s
Reset	1.078s	1.067s	1.075s	1.107s	1.002s	1.066s

5. Discussion

Based on the review of previous literature, the main takeaway is that passwords cannot be fully replaced although it comes with various issues and limitations, passwords are the most cost-effective and usable authentication mechanism so far. Besides that, the majority of the conducted survey agree that passwords are a good security mechanism for their digital security and privacy while conversely, the same majority agrees that password management is hard for them. According to the analyzed results, it indicates that malpractices and lack of best security practices can be seen even among the well-educated and computer-literate communities showcasing the fact that manual password management and semi-automated password management approaches are not feasible enough to provide better password management with high security and considerable usability though it was strongly recommended as stated by Grilo, Ferreira and Almeida [19]. Despite that, it reveals the necessity of an automated solution for password management which takes off the burden of password management from the computer users whom we consider the weakest link of the computer security chain as claimed by Naing Oo [6].

It was identified that the majority of users are reluctant to utilize both offline and online software-based password managers due to potential security threats and vulnerabilities that they might cause over time. Therefore, the majority would rely on manual password creation and practicing memorization as the password storage mechanism. Hence, there is a high rate of password reuse and a low rate of password renewal even among users who are well-exposed to computer technology (Table 4).

Table 4. Survey analysis on malpractices and password management

Malpractice	Password creation method		Password saving method			
	Manual 130	Generated 35	Memorization 63	Physical & digital notes 56	Password managers 44	Hardware token 2
Reuse	102	23	55	49	34	0
Not renewing	75	26	31	32	29	2

Similar to the findings of Gao et al. [3], Stobert and Biddle [21] and Pearman et al. [4], survey analysis also indicates that the growing number of password-protected accounts increases the number of passwords that a user has to memorize over time and utilizing memorization as the primary storage mechanism increases the reuse of the same password or same set of passwords over multiple accounts due to the limited brain memory capacity. Therefore, users are reluctant to practice periodic password renewal to avoid confusion between old and new passwords. Survey result analysis also indicates that manual password creation would lead users to create very simple passwords (e.g.: - name, mobile number) to easily guessable passwords within the context of the user and password-protected account's environment (e.g.: - school name, admission number) which has been also supplementarily reported by Glory et al. [14].

In addition to the findings of Chaudhary et al. [5], the utilization of password managers can be seen at a lower rate as the majority of them are cloud-based password managers with the majority unaware of hardware password wallets. Despite the facts reported by Anand, Susila and Balakrishnan [32], Gupta et al. [33] and Pearman et al. [4], users who are troubled with the difficulty and insecurity of password management and software-based password managers are willing to sacrifice the convenience of cloud-based password managers to get an automated solution and bear the burden of carrying a small device like smartphone or smartwatch as their password wallet. This survey outcome is consistent with the outcome of a survey conducted by Aebischer et al. [35], which not only aligns with the idea of password management being automated but also highlights the willingness to carry a physical password wallet. The analysis agrees with the fact that security literate users only trust that they have a satisfactory level of control over their data and related operations as stated by Wang and Khan [34].

Table 5. Feature comparison matrix of password managers

PM	Base	Opensource	Password storage	Password generation	Mode	Reuse mitigation	Periodic renewal	Automated
Pico	Hardware	Yes	No	No	Offline	No	No	No
PUF based	Hardware	Partial	No	Yes	Offline	No	No	No
E2PM	Hardware	Yes	Yes	No	Offline	No	No	No
DashLane	Software	No	Yes	Yes	Cloud	No	No	Partial
LastPass	Software	No	Yes	Yes	Cloud	No	No	Partial
KeePass	Software	No	Yes	Yes	Offline	No	No	Partial
SecretCentric	Hardware	No	Yes	Yes	Offline	Yes	Yes	Yes

Both Pico and PUF-based solutions are hardware password managers that offer certain advantages, such as offline usage and open-source availability while they lack critical features such as password storage, renewal, and the ability to mitigate password reuse. Although E2PM supports password storage and open-source availability, it lacks password generation and automated processes. *SecretCentric*, on the other hand, encompasses all these functionalities, providing a comprehensive solution that caters to users' needs for secure password management and automates the whole process. Moving on to software-based password managers, we evaluated three popular options: Dashlane, LastPass, and KeePass. These software solutions offer cloud-based storage and password generation capabilities, but they share a common limitation of not being able to mitigate password reuse effectively. Furthermore, they lack automated features like periodic password renewal, which can significantly enhance overall password security. *SecretCentric*, being a hardware-based solution, eliminates the risk of cloud-based breaches and online attacks.

Comparing *SecretCentric* with existing solutions, it is evident that *SecretCentric* provides a unique and valuable set of features. By combining hardware-based security, offline functionality, password storage, password generation, and advanced automation, *SecretCentric* presents a holistic approach to password management. Its ability to ensure password health by tracking usage and access of user passwords, specifically detecting any instances where they have been utilized on unfamiliar or unrecognized devices sets it apart from other solutions. By periodically renewing passwords, it minimizes the risk associated with long-term password exposure and increases overall security. This proactive approach demonstrates the commitment to maintaining robust password protection. Additionally, the hardware-based design provides an added layer of security compared to software-based alternatives. By keeping passwords offline, *SecretCentric* eliminates the risk of data breaches and unauthorized access through online attacks. This offline approach enhances user trust and confidence in the security of their passwords.

However, *SecretCentric* encounters challenges when it comes to capturing webpage elements in client-side rendered applications and non-standard webpage components. It cannot provide automated processes for web services that do not adhere to standard signup and login forms. Furthermore, *SecretCentric* does not possess the capability to support service URLs that are updated through Content Delivery Networks (CDNs). *SecretCentric's* integration across multiple platforms and operating systems is limited, as it primarily focuses on compatibility with the Windows platform, thereby reducing its accessibility and functionality on non-Windows platforms.

6. Conclusion

When it comes to password management, it always deals with the trade-off between security and usability. Password managers were introduced to address this everlasting password management problem. The problem with software-based password managers is either online managers which store user credentials on third-party cloud storage or offline managers which store user credentials only on a single device. As this paper discussed the significance of the hardware-based password manager solution, it also highlights the importance of automating the process of a password manager to mitigate malpractices of password management such as reuse, non-renewal, and improper storage of passwords. The proposed solution covers password generation, AES-256 encrypted secure communication, single MAC-based password backup, fingerprint authentication, autofill, and device migration functionalities in an automated manner with three major components namely, SC-wallet (a secure password vault), SC-client (a client-side desktop application), and SC-extension (a dedicated web browser extension). This proposed methodology allows users a password-free computer technology experience without hassling password management anymore.

6.1. Limitation

The main drawback of the proposed solution is the burden of carrying the SC-wallet everywhere in case of the need to get authenticated for a password-protected account. Additionally, the SC-client and SC-extension must be installed on user devices before using the *SecretCentric* solution with SC-wallet. Although the proposed solution is enriched with a theft and loss-proof mechanism, users may have to go through a rough time re-configuring a virgin SC-wallet. Furthermore, the proposed solution provides the security of credentials up to browser autofill only. Beyond that,

SecretCentric cannot guarantee the security of both credentials and users.

6.2. Future Work

Future research will focus on improving *SecretCentric's* usability by assessing formal metrics such as setup time, learning curve, and user satisfaction. Scalability testing under high loads on resource-constrained hardware, like the Raspberry Pi, will also be conducted to validate its performance. Empirical benchmarking and quantitative comparisons with existing solutions will substantiate the system's theoretical advantages. Additionally, to address potential adoption challenges, future iterations will explore more portable and user-friendly form factors, such as keychain or wallet integration, while user studies will provide deeper insights into preferences and adoption barriers.

References

[1] V. Taneski, M. Heričko, and B. Brumen, "Password security — No change in 35 years?," *IEEE Xplore*, 2014. https://ieeexplore.ieee.org/document/6859779.

[2] N. Sebastian, "Top Password Strengths and Vulnerabilities: Threats, Preventive Measures, and Recoveries," *www.goodfirms.co*, 2021. https://www.goodfirms.co/resources/top-password-strengths-and-vulnerabilities#:~:text=30%25%20of%20the%20Users%20Have

[3] X. Gao, Y. Yang, C. Liu, C. Mitropoulos, J. Lindqvist, and A. Oulasvirta, "Forgetting of Passwords: Ecological Theory and Data Forgetting of Passwords: Ecological Theory and Data," 2018.

[4] S. Pearman, S. Zhang, L. Bauer, N. Christin, and L. Cranor, "Open access to the Proceedings of the Fifteenth Symposium on Usable Privacy and Security is sponsored by USENIX. Why people (don't) use password managers effectively Why people (don't) use password managers effectively," 2019.

[5] S. Chaudhary, T. Schafeitel-Tähtinen, M. Helenius, and E. Berki, "Usability, security and trust in password managers: A quest for user-centric properties and features," *Comput. Sci. Rev.*, vol. 33, pp. 69–90, 2019, doi: 10.1016/j.cosrev.2019.03.002.

[6] A. Naing Oo, "E2PM: Enclosed Portable Password Manager," 2022.

[7] R. Morris and K. Thompson, "Password security: a case history," *Commun. ACM*, vol. 22, no. 11, pp. 594–597, 1979, doi: 10.1145/359168.359172.

[8] M. Yıldırım and I. Mackie, "Encouraging users to improve password security and memorability," *Int. J. Inf. Secur.*, vol. 18, no. 6, pp. 741–759, 2019, doi: 10.1007/s10207-019-00429-y.

[9] H. Habib *et al.*, "User Behaviors and Attitudes Under Password Expiration Policies," *www.usenix.org*, 2018. https://www.usenix.org/conference/soups2018/presentation/habib-password

[10] S. Lyastani, M. Schilling, S. Fahl, M. Backes, and S. Bugiel, "Better managed than memorized? Studying the Impact of Managers on Password Strength and Reuse," 2018.

[11] D. Fredericks, "Users' Perceptions Regarding Password Policies," 2018.

[12] C. Rahalkar and D. Gujar, "A Secure Password Manager," *Int. J. Comput. Appl.*, vol. 178, no. 44, pp. 5–9, 2019, doi: 10.5120/ijca2019919323.

[13] C. Luevanos, J. Elizarraras, K. Hirschi, and J. Yeh, "Analysis on the Security and Use of Password Managers," *2017 18th Int. Conf. Parallel Distrib. Comput. Appl. Technol. PDCAT*, 2017, doi: 10.1109/pdcat.2017.00013.

[14] F. Z. Glory, A. Ul Aftab, O. Tremblay-Savard, and N. Mohammed, "Strong Password Generation Based On User Inputs," *2019 IEEE 10th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON*, 2019, doi: 10.1109/iemcon.2019.8936178.

[15] N. Woods and M. Siponen, "Improving password memorability, while not inconveniencing the user," *Int. J. Hum.-Comput. Stud.*, vol. 128, no. 128, pp. 61–71, 2019, doi: 10.1016/j.ijhcs.2019.02.003.

[16] A. Singh and S. Raj, "Securing password using dynamic password policy generator algorithm," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1357–1361, 2022, doi: 10.1016/j.jksuci.2019.06.006.

[17] D. Biesner, K. Cvejoski, B. Georgiev, R. Sifa, and E. Krupicka, "Generative Deep Learning Techniques for Password Generation," *ArXiv201205685 Cs*, 2020, Accessed: Jan. 01, 2022. [Online]. Available: http://arxiv.org/abs/2012.05685

[18] K. Smith, "Random Password Generation," 2022. Accessed: Jan. 01, 2022. [Online]. Available: https://covacci.org/wp-content/uploads/2022/04/Kirk-Smith_Random-Password-Generation.pdf

[19] M. Grilo, J. F. Ferreira, and J. B. Almeida, "Towards Formal Verification of Password Generation Algorithms used in Password Managers," *ArXiv210603626 Cs*, 2021, Accessed: Jan. 01, 2022. [Online]. Available: https://arxiv.org/abs/2106.03626

[20] E. Kuka and R. Bahiti, "Information Security Management: Password Security Issues," *Acad. J. Interdiscip. Stud.*, vol. 7, no. 2, pp. 43–47, 2018, doi: 10.2478/ajis-2018-0045.

[21] E. Stobert and R. Biddle, "The Password Life Cycle," *ACM Trans. Priv. Secur.*, vol. 21, no. 3, pp. 1–32, 2018, doi: 10.1145/3183341.

[22] E. I. Tatli and E. Seker, "Password Replacement Patterns," *2018 5th Int. Conf. Control Decis. Inf. Technol. CoDIT*, 2018, doi: 10.1109/codit.2018.8394966.

[23] M. Abuzaraida and A. Zeki, "Collection of Handwritten text View project Development of Malay Online Virtual Integrated Corpus (MOVIC) for Sentiment Analysis using Web-scraping View project AWARENESS AND SECURITY ISSUES IN PASSWORD MANAGEMENT AMONG LIBYAN UNIVERSITIES STAFF MEMBERS," *Artic. ID IJARET1112123 Int. J. Adv. Res. Eng. Technol.*, vol. 11, no. 12, pp. 1292–1303, 2020, doi: 10.34218/IJARET.11.12.2020.123.

[24] K. Siddique, Z. Akhtar, and Y. Kim, "Biometrics vs passwords: a modern version of the tortoise and the hare," *Comput. Fraud Secur.*, vol. 2017, no. 1, pp. 13–17, 2017, doi: 10.1016/s1361-3723(17)30007-6.

[25] V. Zimmermann and N. Gerber, "The password is dead, long live the password – A laboratory study on user perceptions of authentication schemes," *Int. J. Hum.-Comput. Stud.*, vol. 133, pp. 26–44, 2020, doi: 10.1016/j.ijhcs.2019.08.006.

[26] D. E. Kurniawan, M. Iqbal, J. Friadi, F. Hidayat, and R. D. Permatasari, "Login Security Using One Time Password (OTP) Application with Encryption Algorithm Performance," *J. Phys. Conf. Ser.*, vol. 1783, p. 012041, 2021, doi: 10.1088/1742-6596/1783/1/012041.

[27] S. Kankane, C. DiRusso, and C. Buckley, "Can We Nudge Users Toward Better Password Management?," *Ext. Abstr. 2018 CHI Conf. Hum. Factors Comput. Syst.*, 2018, doi: 10.1145/3170427.3188689.
[28] R. Macgregor, "USER COMPREHENSION OF PASSWORD REUSE RISKS AND MITIGATIONS IN PASSWORD MANAGERS," 2020.
[29] P. Gupta, D. R. Marur, H. Kalisetty, and A. Khanna, "A novel secure and high-entropy hardware password manager," *Mater. Today Proc.*, 2020, doi: 10.1016/j.matpr.2020.09.524.
[30] Q. Guo *et al.*, "PUFPass: A password management mechanism based on software/hardware codesign," *Integration*, vol. 64, pp. 173–183, 2019, doi: 10.1016/j.vlsi.2018.10.003.
[31] P. Sabev and M. Petrov, "Android Password Managers and Vault Applications: An Investigation on Data Remanence in Main Memory," *Inf. Syst. Grid Technol.*, vol. 2933, pp. 314–328, 2021.
[32] S. Anand, N. Susila, and S. Balakrishnan, "Challenges and issues in ensuring safe cloud based password management to enhance security," 2018.
[33] A. Gupta, A. Sahu, A. Tarodia, and S. Choudhari, "SeCrypt : A Password Manager Aniket Sahu 3 PUBLICATIONS 0 CITATIONS SEE PROFILE," *Artic. Int. J. Innov. Res. Sci. Eng. Technol.*, 2022, doi: 10.15680/IJIRSET.2022.1105125|.
[34] Y. Wang and K. M. Khan, "Matrix Barcode Based Secure Authentication without Trusting Third Party," *IT Prof.*, vol. 21, no. 3, pp. 41–48, 2019, doi: 10.1109/mitp.2018.2876986.
[35] S. Aebischer *et al.*, "Pico in the Wild: Replacing Passwords, One Site at a Time," *Proc. 2nd Eur. Workshop Usable Secur.*, 2017, doi: 10.14722/eurousec.2017.23017.
[36] F. Stajano, "Pico: No More Passwords!," *Secur. Protoc. XIX*, pp. 49–81, 2011, doi: 10.1007/978-3-642-25867-1_6.
[37] M. Mohammadinodoushan, B. Cambou, F. Afghah, C. R. Philabaum, and I. Burke, "Reliable, Secure, and Efficient Hardware Implementation of Password Manager System Using SRAM PUF," *IEEE Access*, vol. 9, pp. 155711–155725, 2021, doi: 10.1109/access.2021.3129499.
[38] M. Mohammadinodoushan, B. Cambou, C. R. Philabaum, and N. Duan, "Resilient Password Manager Using Physical Unclonable Functions," *IEEE Access*, vol. 9, pp. 17060–17070, 2021, doi: 10.1109/access.2021.3053307.

Authors' Profiles

Prageeth Fernando holds a Bachelor's degree in Information and Communication Technology with a specialization in Networking Technology from the University of Sri Jayewardenepura and is currently pursuing an MSc in Cybersecurity at the Sri Lanka Institute of Information Technology. His expertise spans computer networking, cyber security, and cryptography, with research interests focused on secure user authentication systems, advanced cryptographic techniques and integration of ICT in education.

Navinda Dissanayake is a Bachelor of Information and Communication Technology graduate from the University of Sri Jayewardenepura. His areas of interest include web development, software engineering, and database programming. Navinda has co-authored research on implementing intelligent attendance systems and improving user authentication methods.

Venusha Dushmantha holds a Bachelor of Information and Communication Technology degree from the University of Sri Jayewardenepura. His expertise lies in object-oriented programming, software development, and web technologies. Venusha has collaborated on research projects aimed at developing smart attendance systems and enhancing password management solutions.

Chamara Liyanage is a senior lecturer at University of Sri Jayewardenepura specializing in machine learning and artificial intelligence. His research encompasses water quality monitoring, smart city planning, and the application of AI in environmental sustainability. Dr. Liyanage has published extensively, contributing to advancements in AI-driven environmental monitoring systems.

Chamila Karunatilake is a lecturer at University of Sri Jayewardenepura and his research focuses on music information retrieval, multimedia data processing, and the application of machine learning in audio analysis. He has explored traditional and folk melody classification using Markov models and neural networks, contributing to the field of computational musicology.

How to cite this paper: Prageeth Fernando, Navinda Dissanayake, Venusha Dushmantha, Chamara Liyanage, Chamila Karunatilake, "An Automated Solution for Secure Password Management", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.10, No.4, pp. 21-33, 2024. DOI: 10.5815/ijmsc.2024.04.03