VaultSGX : A Secured Password Managing System with Intel SGX

Karan Tejwani *kt9576*
Pranjali Thakur *pt4202*
Vyshnavi Maddela *vm2429*

Abstract

Protecting sensitive user data passwords is a top priority in today's digital environment. The need for efficient password management systems is highlighted by the challenges of creating strong passwords for a variety of applications, the difficulty of memorizing passwords, and being susceptible to unauthorized access. The main objective of this project is to develop and build a secure password manager using Intel Software Guard Extension (SGX) technology to meet this critical requirement. The project includes methods for user authentication using master passwords that are utilized in a secure enclave. The fundamental element of this approach is the construction of a secure vault inside the enclave, which functions as an isolated space set aside for password management and storage to thwart unwanted access even in the case of a compromised host system.

Structured into essential stages, the Initialization Phase safely constructs the password vault, laying the groundwork. When user credentials are encrypted and sealed inside the SGX enclave, the Authentication and Storage Phase enables the addition of entries to the password vault. The secure retrieval and decryption of credentials that are preserved is ensured by the retrieval and decryption phase. The solution, which makes use of SGX enclaves, is focused on safeguarding confidential data by employing encrypted procedures in a secure, isolated environment. Before transmitting the passwords to the applications that request it, every password is re-encrypted using keys unique to the service to further strengthen the system. Furthermore, the system has secure wiping processes that eliminate all evidence of confidential information and reduce the possibility of unintentional disclosure. By combining cutting-edge cryptographic methods with strong user authentication and enclave features, this combination creates a password management system that is immune to changing cybersecurity threats and guarantees the confidentiality of users' assets and identities in a constantly shifting digital environment.

1 Introduction

In a period when internet connectivity is almost universal, protecting user passwords and personal data has become critical. The growing number of online services that demand strong, one-of-a-kind passwords has made sophisticated password management systems imperative. In response to user concerns about creating strong, secure passwords, this project presents VaultSGX, a cutting-edge password manager strengthened by Intel Software Guard Extensions (SGX) technology.

The deficiencies of conventional password management solutions in the face of growing cybersecurity risks are the driving force behind VaultSGX. By utilizing the hardware-based security capabilities offered by Intel SGX, the initiative seeks to completely rethink the security paradigm [12]. VaultSGX seems as a complete solution meant to not only improve password security but also completely transform the password management process as users struggle with the difficulties of coming up with and maintaining strong passwords for a variety of applications.

Since users have numerous passwords to come up with and remember for different applications, they frequently turn to methods that unintentionally risk the security of their personal information. Users are exposed to several risks by using common password management techniques, such as storing passwords in plain sight, using the same password on multiple devices or applications, or neglecting regular password changes because they are hard to remember [17]. The widespread practice of keeping passwords in an easily accessible format is one obvious drawback. Many people underestimate the risks that are involved with publicly keeping passwords in text files or even on physical notes, stating ease as their reason. In the case of a security breach or the loss of physical notes, this exposes a user's digital identity to unwanted or malicious threat actors [5]. Furthermore, security risks thrive in environments where passwords are rarely changed, a tendency that frequently results from memory difficulties. Using the same password consistently on several different platforms increases the likelihood of a breach of security. A single password

breach might provide intrusion into an individual's private information, creating serious security risks.

Password managers are essential for protecting sensitive login information for users. It is crucial to understand the differences between password managers that store data locally and the one that is created using Intel Software Guard Extensions (SGX) in a secure and trusted environment [2]. Passwords are saved in an encrypted format on the user's device using locally stored password managers, which normally rely on encryption methods and safe storage techniques to protect user data. Although this offers some protection, there are still certain risks that the user's device might face, such as viruses, malware, or sometimes unauthorized access. However, password managers that make use of Intel SGX function in a secure or trusted enclave, hardware-based- trusted execution environment [7]. By establishing isolated, encrypted environments inside the CPU, SGX makes sure that private information kept inside the enclave is shielded from prying eyes even in the event that the device is compromised [1]. When compared to more conventional local storage approaches, this hardware-based technology greatly increases the secrecy of the data that is stored and makes it much more resilient to attacks. Password managers that utilize SGX technology demonstrate a proactive approach to cybersecurity by utilizing hardware-based security features to protect users.

As people find it more and more difficult to set up and recall passwords for a variety of applications, a creative and reliable solution is needed. Using the cutting-edge features of Intel Software Guard Extensions (SGX), this research sets out to tackle these issues head-on and presents a novel method of password management [1]. The project's motivation originates from an awareness of the shortcomings of traditional password management solutions. Although passwords are the defenders of our data and assets, conventional techniques for storing and obtaining them are frequently exposed to various forms of cyberattacks. A proper change in the way we think about and approach the security of passwords is vital in this digital age of numerous information thefts and security breaches. Intel SGX is a technology that allows us to raise the bar for password management security to previously unheard-of levels.

The main goal of this project is to create and deploy a password manager that is strengthened by the strong security features offered by Intel SGX. The project's goal is to build a strong vault for user passwords by isolating critical data inside the secure enclave. The focus is on a wide range of features, such as master password authentication, sealing and unsealing of the credentials, encryption using plugin keys, and secure communication protocols, in addition to reliable storage. When these components are combined, there is comprehensive and strong security against attackers who could try to get user credentials without authorization.

The deliberate selection of Intel SGX as the basis for VaultSGX highlights the dedication to modern security protocols [12]. Secure execution environments are offered by SGX enclaves, which isolate important processes and data from any security breaches. This marks a break from convention by adopting a technology solution that prioritizes user trust, convenience of use, and adaptation to new cybersecurity challenges all while improving password protection.

As we explore the complexities of the project, we will make our way through the several security tiers and see how SGX changes the password management setting. The process includes creating secure passwords, registering users, and allowing programs to easily retrieve credentials all within the protected walls of the SGX enclave. Moreover, the initiative broadens its scope to encompass cutting-edge functionalities such as automated password creation, contributing a level of approachable intricacy to the security landscape.

2 Related Work

To improve safety and performance in enclave-based computing, Intel created Software Guard Extensions 2 (SGX2), an enhancement to the SGX1 programming model. The paper [11] describes the validation procedure for SGX2. Maintaining security-sensitive data across logical processors while optimizing parallelism is a challenging topic the validation addresses. Passwords and other sensitive information are shielded from unwanted access by SGX technology, even in the event of possible security breaches [2]. Password management and storage may be carried out in a highly secure enclave with the help of SGX2's enhanced security capabilities. The password manager's flexibility can be increased by utilizing the dynamic memory allocation supported by the proposed SGX2 [11]. This feature helps the enclave handle variable-sized password databases and increase overall performance by enabling the enclave to dynamically allocate and manage memory resources.

The research [4] offers insightful information on how Intel SGX technology may be used to create safe password managers. The notion of Trusted Execution Environments (TEEs) is examined in this paper along with some of its main advantages, such as confidentiality assurances, hardware-based security, and remote attestation. It goes into further detail on Intel SGX, providing technical information on how to create enclaves, data security measures, and accessible APIs. Password manager developers may employ SGX to design highly secure systems that shield sensitive user credentials from manipulation and illegal access by comprehending these factors. The research [4] also looks at SGX implementation in cloud contexts, which opens up opportunities for improved cloud-based password storage and safe multi-party calculations.

The study [5] uses Intel SGX technology to propose a unique password file security solution. This article provides a tangible implementation example, UniSGX, built on the theoretical foundations of TEEs and SGX. Within the PAM authentication system, UniSGX shows how well SGX en-

claves safeguard critical credentials. UniSGX improves the overall security posture of the authentication process and reduces possible vulnerabilities by using enclaves for password management and verification. The study introduces SGX sealing for password file storage, therefore delving deeper into the security benefits of UniSGX. These findings are extremely significant to the creation of secure password managers that make use of Intel SGX. UniSGX is a useful resource for creating and executing reliable password management systems.

The paper [14] provides insightful information that may be used in the development of safe password managers that leverage SGX technology. Its principal contribution is the advancement of hardware-backed isolation, an idea that is essential for safeguarding passwords and other sensitive data from even the most advanced software-based attacks. The study shows how to separate critical processes and data from the untrusted environment by utilizing SGX enclaves, providing a degree of security that is not possible with conventional software-based solutions. This approach is useful for building a strong tamper-resistant password manager and ensures integrity and confidentiality against software and physical attacks. When compared to current methods, the paper's [14] two-factor authentication system within enclaves, for example, provides improved security against unauthorized access and brute-force attacks.

Robust security solutions are required since standard password databases are inherently vulnerable to offline attacks. The paper [10] identifies this crucial gap and offers a novel solution employing Intel SGX enclaves as a stronghold for password security. Even in compromised systems, passwords are protected from unwanted access and alteration by segregation within enclaves. When compared to offline attack-prone old systems, this greatly improves security. Password-hashing encryption keys are kept and utilized inside enclaves, guaranteeing secure handling and storage. As a result, the security posture of the password manager is further strengthened and attackers are prevented from obtaining these crucial keys. The whole password verification procedure takes place inside enclaves, removing the possibility of manipulation [10]. This ensures that password comparisons are accurate and reliable.

The paper [20] offers a thorough and up-to-date analysis of SGX technology, its features, and its potential for safeguarding sensitive information that is pertinent for the creation of secure password managers. The study provides a strong foundation for understanding how SGX may be used to safeguard user credentials against unwanted access and modification by exploring some of its basic features, including memory isolation and remote attestation. The study's assessment of several SGX applications, including password management, provides insightful information for creating dependable and safe password management systems. In comparison to conventional software-based solutions, the article demonstrates how SGX may greatly improve the security posture of password managers by emphasizing the advantages of enclave-based password processing and storage [20].

A strong argument is made in the paper [16] for the use of Intel SGX as a strong security-enhancing tool. Its comprehensive implementation guide, which focuses on C/C++ code integration, is an invaluable blueprint for creating safe password processing and storing SGX enclaves. SGX delivers a hardware-backed layer of security against unwanted access and manipulation by isolating critical data and processes within these enclaves, which far exceeds standard software-based solutions. In addition, the paper's examination of the performance overhead related to SGX offers significant perspectives for enhancing enclave architecture and guaranteeing a seamless user experience.

Strong security measures are required since password database breaches are a constant threat. To address this issue head-on, the paper [3] suggests a novel solution using Intel SGX enclaves to protect critical password data. The document describes in detail how to isolate password storage in SGX enclaves such that it cannot be accessed or altered by unauthorized parties, even in settings that have been hacked, this strengthens the security posture of the password manager. The paper [3] outlines techniques for storing and utilizing encryption keys inside enclaves and highlights the need for safe key management.

The study [6] might serve as a guide for creating safe data transmission features in our password manager. This allows for safe communication across various features, such as processing, authentication, and password storage, all protected behind SGX enclaves and trusted environments. Furthermore, Squad's focus on data isolation inside enclaves offers helpful methods for securing encryption keys and passwords, limiting unwanted access even in compromised settings [6]. When optimizing our password manager for a seamless user experience, it is important to make more informed decisions if you are aware of the possible performance consequences of utilizing SGX.

The Memory Encryption Engine (MEE), a hardware module for encrypting and decrypting system memory on general-purpose CPUs, is proposed in the paper [8]. The transparent hardware-accelerated encryption offered by MEE can give password data kept in SGX enclaves an extra degree of security. Passwords and other sensitive data would be encrypted in memory even if the enclave was breached, drastically minimizing the possibility of unwanted access. One may selectively secure particular memory areas within the enclave through MEE's granular encryption management, making sure that the most important data is protected to the fullest degree possible.

Due to their dependence on untrusted memory for data processing and storage, traditional database systems are vulnerable to attacks such as code injection and memory scraping. By processing and storing sensitive data inside SGX enclaves, the paper EnclaveDB [13] addresses these vulnerabilities by establishing an environment of security that is

impenetrable by outside interference and manipulation. Assuring data consistency and integrity even in the event of crashes or attacks, EnclaveDB incorporates a secure transaction manager within enclaves. EnclaveDB makes sure one is interacting with a legitimate and secure enclave by validating the SGX environment's integrity before permitting any contact [13]. Protecting user passwords is precisely aligned with EnclaveDB's main idea of enclave-based data processing and storage. Even in compromised environments, sensitive data, such as encryption keys and login passwords, remain protected within the safe boundaries of enclaves, making it resilient to manipulation and attacks. It guarantees that all password-related activities from retrieval to modification occur completely inside the enclave.

The study [18] provides an in-depth analysis of current applications that make use of Intel SGX technology, emphasizing how these applications may improve data privacy, confidentiality, and integrity. The fundamental idea behind SGX is that it offers isolated execution environments, or enclaves, which shield data and code from unwanted access even when a system is hacked. The study [18]examines more than 290 SGX-enabled applications across a variety of fields, including secure enclaves for cryptocurrency wallets, medical record storage, and password managers, among others which draws attention to secure enclaves as the essential component of password management provided by SGX. Solutions that already exist that use enclaves to store, retrieve, and manipulate passwords offer practical examples and methods that we may modify for our project.

Intel Software Guard Extensions (SGX) is a strong defender in the world of secure enclaves, emphasizing enclave security and confidentiality foremost. By segregating apps into trusted and untrusted parts, SGX creates an environment for private information [2]. Diverse technologies for secure enclaves have been created to prevent unwanted access to confidential information and explain how it may be used to secure passwords. Fundamentally, the ability to construct "protected enclaves"—isolated execution environments within the processor itself—is what SGX offers developers [2]. These protected areas serve as safe zones for private information, protecting passwords from theft, alteration, and even possible operating system breaches. Because of the substantial reduction in attack surface caused by this isolation, it is extremely difficult for bad actors to break the digital fortress guarding user credentials. The implementation of Intel SGX in this project is notable for its hardware-based approach to guaranteeing the secrecy and integrity of enclaved data.

By utilizing the features of Intel SGX enclaves, the article [9] offers an effective solution to the critical problem of web password security. Understanding the inherent vulnerabilities in conventional web authentication, SafeKeeper offers a revolutionary solution that separates the processing of critical passwords into hardware-backed enclaves, providing an additional layer of defense against server breaches and phishing attempts even in compromised settings. This strategy offers a plan for putting secure enclave-based password processing, authentication, and storage into practice, which is exactly in line with the main goal of our password manager project. This system employs a server-side Trusted Execution Environment to protect web passwords, defending against phishing and password database theft. Furthermore, the paper's focus on user-friendliness via browser extensions and visual signals encourages sensible password management techniques and increases user acceptance.

The article [19] describes SGX-UAM, a brand-new Unified Access Management (UAM) system that makes use of one-time passwords (OTPs) and Intel Software Guard Extensions (SGX). It uses SGX enclaves to process and store user credentials securely and dramatically lowers the attack surface by isolating important data within these protected memory areas. One further layer of security is added by integrating one-time passwords (OTPs) created inside of enclaves, which reduces the exposure of real credentials during logins and other sensitive actions. The paper [19] shows how SGX may be used to create password managers that prioritize user privacy and reduce needless data exposure by keeping user data contained within enclaves. SGX-UAM prioritizes security and guards against client attacks, Man-in-the-Middle attacks, phishing, replay, and Denial of Service attacks. Notably, it lowers costs by introducing a lightweight OTP solution without requiring extra hardware [19]. This explains the threats related to password manger and shows the ways to tackle them.

Using safe hashing, PwdHash creates a new password for every website, reducing the possibility of using the same login information and guarding against phishing scams. The paper's [15] emphasis on user ease and seamless browser integration is in line with the demand for password management applications to have intuitive user interfaces. However, it's malware susceptibility and dependence on browser security show how limited browser-based solutions are in comparison to SGX's hardware-enforced isolation [15]. We obtain a more comprehensive picture of the varied landscape of password security systems by analyzing PwdHash in conjunction with SGX-based methods. By contrasting and comparing their features, security advantages and disadvantages, and user experience concerns, we can finally lead the design of a reliable and approachable password manager that makes use of the best aspects of both worlds.

Although Intel SGX enclaves appear to provide an unbreakable barrier for processing and storing passwords, the threat of cache attacks remains. "Cache Attacks on Intel SGX" [7] sheds light on this issue by demonstrating how attackers might extract sensitive data, perhaps including user passwords, by using side-channel information obtained from shared caches. This finding emphasizes the necessity of using several security layers in an SGX-based password manager, even when the enclave's walls appear to be safe. The research [7] shows how memory access patterns within the enclave may be found by

attackers using strategies like Flush+Reload and Prime+Probe, which might lead to the reconstruction of sensitive data. This underscores the intrinsic constraints of isolation based on hardware and stresses the significance of countermeasures based on software. Countermeasures like constant-time operations and memory randomization become essential for your SGX password manager.

3 Adversary Model

In the ever-changing field of cybersecurity, protecting user data especially confidential data like passwords from attackers is a constant challenge. User credentials are protected by the password manager, which has been strengthened by Intel's Software Guard Extensions (SGX) technology. However, even the strongest defenses might have weaknesses and vulnerabilities. Therefore, it is essential to comprehend the capabilities of prospective attackers while creating an effective password manager. The adversary model in this project's setting includes a range of attack scenarios that might compromise the integrity and confidentiality of user credentials kept inside the Intel SGX enclave. The initiative is to counter these possible attacks and strengthen the system's security against attackers.

The main goal of the attacker is to get or breach user credentials that are stored in the SGX enclave of the VaultSGX password manager. The adversary's additional goals, include obtaining unauthorized access to sensitive user data, tampering with the password manager's operation, and compromising the integrity and secrecy of the passwords that are stored. These diverse objectives draw attention to how broad the potential risks are, highlighting the necessity of strong security that will protect the privacy and integrity of data of the user inside the password manager's protected enclaves.

Having hardware access to the host machine running VaultSGX gives the attacker a significant advantage since it allows them to potentially manipulate important hardware components and take control of the operating system. With this degree of access, an attacker might possibly compromise the integrity of the entire system by exploring hardware architectural vulnerabilities. A significant threat is posed by the adversary's ability to change hardware components, which allows them to take advantage of potential gaps in the system's foundation. In addition, their control over the operating system indicates they can circumvent and bypass security protocols, which increases the likelihood of unwanted access and possible compromise of private information kept in the password manager's Secure SGX enclave.

Apart from its hardware abilities, the adversary also possesses significant software expertise. They may use intricate tools that include side-channel attacks, buffer overflow exploits, and social engineering techniques. To obtain sensitive data, side-channel attacks take advantage of vulnerabilities in the way cryptographic algorithms are physically implemented. Contrarily, buffer overflows involve making use of programming vulnerabilities to exceed the amount of memory allotted, which may result in the execution of malicious code. The use of social engineering techniques highlights the adversary's versatility even more because it allows them to take advantage of user behavior and trick individuals into divulging their passwords or gaining illegal access.

Attack vectors targeting VaultSGX include a variety of strategies, such as side-channel vulnerabilities and enclave bug exploitation in direct enclave attacks. While side-channel exploits make use of vulnerabilities in the hardware to obtain information illegally, enclave bugs find and exploit vulnerabilities in the SGX enclave's code to possibly get around safety measures. In addition, the adversary uses host system breach strategies, manipulating the behavior of the password manager and taking advantage of operating system vulnerabilities. When it comes to user authentication, the adversary uses keystroke logging to record user keystrokes during login or password management processes, and phishing attacks to trick users into disclosing critical information such as master passwords.

Through memory attacks or memory corruption vulnerabilities, adversaries may try to access confidential information from the enclaves. Through these techniques, attackers might be able to get encryption keys or plaintext passwords from the enclave's memory. The project will use strong programming standards to lower the risk of memory corruption attacks and secure wiping procedures to ensure that sensitive information is immediately and completely deleted from memory after usage to avoid these threats. The enclave should also use hashing and encryption techniques to safeguard private information from efforts at memory corruption.

During the authentication process, adversaries may pretend to be valid users in an effort to get credentials. The project will place a strong user authentication system that uses the master password to prevent illegal access to mitigate this threat. Efforts for user awareness and education must additionally emphasize the significance of credentials being confidential and spot any kind of phishing attempts.

An adversary might compromise the system's security as a whole if they manage to obtain the user's master password. It is important to understand the need for safe master password management procedures, such as hashing and encryption, to reduce this vulnerability. Initiatives for user education should also stress how important it is to protect login credentials and identify potential risks.

In a brute-forcing attack, attackers methodically try every combination in an effort to guess a user's password. The project creates delays between login attempts or employs a strong account lockout mechanism to address this issue. By restricting the amount of successive or total login attempts, limited retry rules prevent brute force attacks by making it difficult for adversaries to guess passwords all the way through.

Adversaries can try to intercept or alter communication be-

tween the non-secure application and the enclave. Therefore. use of secure communication protocols like TLS/SSL and validating the reliability and legitimacy of communication channels can lessen these risks.

The project aims to develop a password management system that is resistant to many forms of attacks by identifying and mitigating these possible adversary situations. Secure coding techniques, user education, and cryptographic protections are all part of a holistic defensive plan that creates a strong security posture against both common and sophisticated adversaries.

4 VaultSGX

Our approach involves a step-by-step implementation of key functionalities, including the creation of a secure enclave for password storage, implementing robust encryption mechanisms, and developing a user-friendly interface for easy interaction with the system. The project implementation can be divided into phases, each focusing on a specific feature or functionality: Initialization Phase, Authentication and Storage Phase, Retrieval and Decryption Phase.

4.1 Initialization Phase: 'Create'

In the Initialization Phase, our primary goal was to establish the foundation of the password management system. This has been successfully achieved by allowing users to create their password wallet. The implemented key functionality includes an input field for the master password. On the backend, the application initializes the wallet with these credentials, ensuring they are securely stored.

The master password entered by the user is encrypted and sealed using Intel SGX's enclave technology. This process is crucial for protecting the credentials against unauthorized access and attacks. The SGX enclave provides a trusted execution environment, where the sensitive data is not only encrypted but also securely stored, ensuring confidentiality and integrity.

The success of this phase has laid a robust framework for the application, emphasizing security and user trust from the outset. We have successfully integrated the SGX SDK to leverage the secure enclave technology, as demonstrated in the Implementation section. Figure 1 illustrates the workflow of this phase.

4.2 Authentication and Storage Phase: 'Add'

The Authentication and Storage Phase focuses on securely adding new entries to the user's password wallet. Following the successful implementation, after user authentication using their master password, the system can now add the details of a new entry, including the service URL, username, and password.

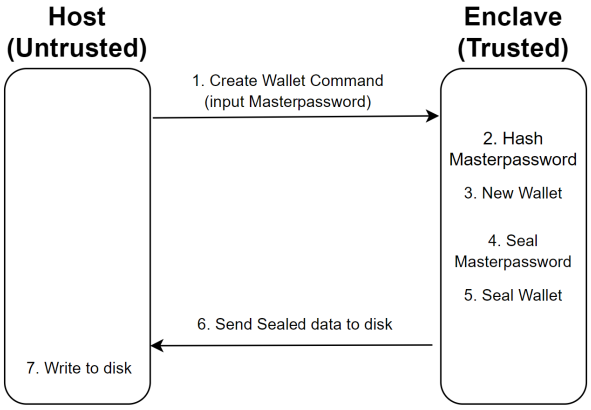

Figure 1: Initialization workflow

This information is then securely sealed within the Intel SGX enclave, ensuring that the data is encrypted with the master password. The encrypted data is stored in a protected area within the enclave, inaccessible to unauthorized users or processes.

This phase highlights our project's intent of strong security by utilizing Intel SGX to protect user data from potential breaches. The process from authentication to secure storage is visually depicted in the accompanying Figure 2.

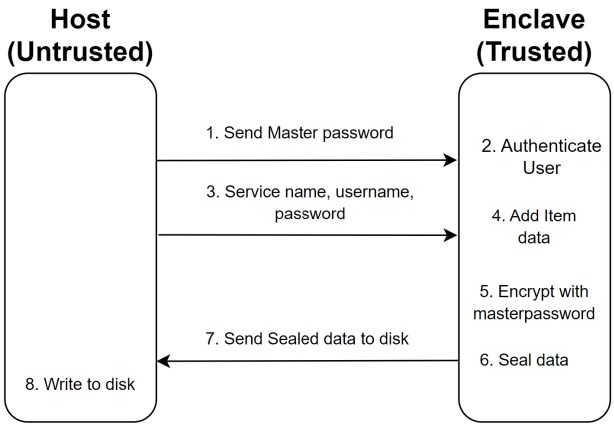

Figure 2: Authentication and Storage workflow

4.3 Retrieval and Decryption Phase: 'Get'

The Retrieval and Decryption Phase is centered on securely fetching and decrypting stored credentials. We have implemented a robust process for unsealing the data stored within the Intel SGX enclave. Upon accessing the data, the system is

capable of fetching the required service name and decrypting the credentials using the master password.

To ensure the highest security standards, the decrypted data is then re-encrypted before being transmitted or used, maintaining confidentiality even in transit. This phase is critical as it involves direct handling of sensitive user data, necessitating stringent security measures to protect against unauthorized access or leaks. The decryption and re-encryption workflow is illustrated in the provided Figure 3.

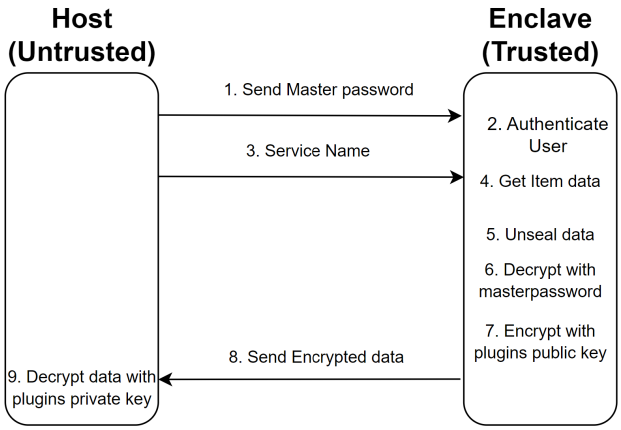

Figure 3: Retrieval and Decryption workflow

The Algorithm 1 shows the systematic representation of the application's core functionality, detailing the step-by-step procedures involved in initializing, authenticating, and managing user credentials.

Figure 4 comprehensive flowchart illustrating the workflow of our SGX-based password manager. This diagram provides a visual representation of the various stages and functions involved in the system's operation, offering a clear understanding of the process flow from initialization to termination.

Algorithm 1 SGX-based Password Manager

Initialize the SGX Enclave.
Check for the existence of sealed_data.bin.
if sealed_data.bin does not exist **then**
    Prompt user to create a master password.
    Encrypt and seal the master password.
    Store the sealed master password in sealed_data.bin.
else
    Prompt user for master password.
    Authenticate the master password.
    if authentication fails **then**
        Delete sealed_data.bin after 3 unsuccessful attempts.
    else
        Offer options to 'Store' or 'Retrieve' credentials.
    end if
end if
if option 'Store' is selected **then**
    Collect service URL, username, and password.
    Encrypt and seal the credentials.
    Store the sealed credentials.
else if option 'Retrieve' is selected **then**
    Prompt for service name.
    Retrieve and unseal credentials for the service.
    Display the credentials to the user.
end if
Terminate the SGX Enclave.

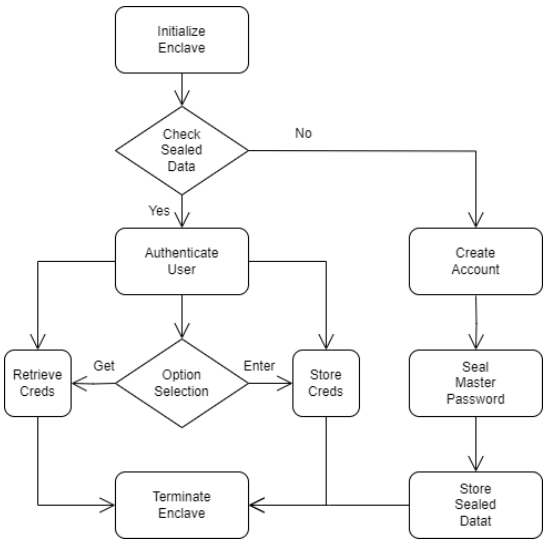

Figure 4: Flowchart

5 Implementation

5.1 Initialization and Account Creation

Running the Application (App.cpp): When the application (./app) is launched, it first checks for the existence of a sealed file in the project repository. If this file does not exist, it indicates that no account has been set up yet. Consequently, the user is prompted to create an account. This is a crucial step for first-time users. The user is then asked to enter a master password. This master password is the primary key for accessing all stored credentials.

Master Password Sealing (App.cpp & Sealing.cpp): The user-entered master password is then passed to the seal function within the SGX enclave. This function, defined in Sealing.cpp, uses SGX's sealing capabilities to encrypt and seal the password. The sealed password data is then written back to the sealed file, securing it outside the enclave.

```
int main(int argc, char const *argv[]) {
    if (initialize_enclave(&global_eid, "enclave.token", "enclave.signed.so") < 0) {
        std::cout << "Fail to initialize enclave." << std::endl;
        return 1;
    }


    sgx_status_t status, retval;

    int ret;
    ecall_get_account(global_eid, &ret);
    std::ifstream sealed_file("sealed_data.bin", std::ios::in | std::ios::binary);
    if (ret == 1) {

    } else {
        // std::cerr << "Sealed data file not found, a new master password must be set." << std::endl;
        // File not found, prompt the user to set a new master password
        std::string new_master_password;
        std::cout << "No master password set. Please set a new master password: ";
        std::cin >> new_master_password;
```

Figure 5: Code for Initialization of Application

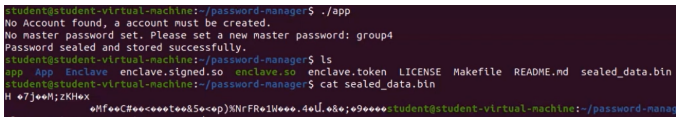

Figure 6: Initialization and Account Creation

5.2 Authentication and Access

Master Password Authentication (Enclave.cpp): If the sealed_data.bin file is present during the application launch, it implies that an account already exists. The user is then prompted to enter the master password. This step is crucial for authenticating the user's identity and ensuring that access to the stored credentials is granted only to the legitimate user. The ecall_auth function in Enclave.cpp prompts the user to enter the master password. This password is then verified against the unsealed data from sealed file.

Action Selection (App.cpp): Upon successful authentication, the user is presented with options to either "Enter" new credentials or "Get" existing credentials. This choice determines whether the user wants to store new credentials or retrieve existing ones.

```
void ecall_auth(const char* master_pass, size_t master_pass_size){
    int res;
    bool authenticated = false;
    int failed_attempts = 0;
    char password_buffer[1024];
    int result;

    do
    {
       ocall_print("Please enter your master password: ");
       ocall_get_string(&res, password_buffer, sizeof(password_buffer));
     if (res == 0) {
       ecall_verify_master_password(password_buffer, strlen(password_buffer), master_pass,
strlen(password_buffer), &result);
      if(result == 1)
      {
        ocall_print("Authentication successful!\n");
        authenticated = true;
        failed_attempts = 0;
      }
      else {
           ocall_print("Authentication failed!\n");
           failed_attempts++;
           if(failed_attempts >= 3)
           {
            ocall_print("Too many failed attempts. Account is deleted.\n");
            ocall_remove_file(sealed_file_path);
            break;
           }
      }
     }

    }
    while(!authenticated);
```

Figure 7: Code for Authentication of Masterpassword

```
void ecall_verify_master_password(const char* input, size_t input_len, const char* actual, size_t
actual_len, int* result) {

    if (memcmp(input, actual, input_len) == 0) {
        *result = 1;
    }
    else {
        *result = 0;
    }
}
```

Figure 8: Code for Verification of Masterpassword

5.3 Credential Management

Storing Credentials: If the user chooses to "Enter", they are asked for the Service Name, Username, and Password. These details are encrypted and stored securely, allowing for secure management and retrieval at a later time.

```
Please enter your master password: group4
Authentication successful!
What would you like to do(Enter or Get)?: Enter
Enter service name: github.com
Enter username: csecgrp4
Enter password: hiall
```

Figure 9: Credential Management

Retrieving Credentials: If the user chooses "Get", they are required to enter the Service Name associated with the cre-

dentials they wish to retrieve. The application then retrieves the corresponding Username and Password. These credentials are decrypted within the enclave before being securely transmitted to the user.

5.4 Security Protocol on Authentication Failure

To enhance security, the application implements a strict protocol for authentication attempts. The ecall_auth function keeps track of failed master password attempts. If the user enters the incorrect password three times, the sealed file is deleted as a security measure to prevent brute force attacks. By deleting the sealed file after multiple failed attempts, the system ensures that the encrypted credentials remain secure.

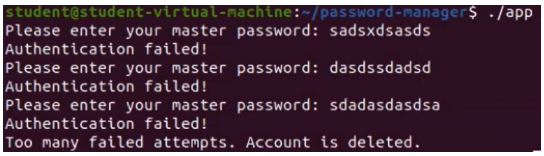

Figure 10: Authentication Failure

Overall, the implementation leverages the security features of Intel SGX to create a secure environment for password management. This involves encrypting and sealing sensitive data like the master password and user credentials, providing robust protection against external threats and system vulnerabilities. The use of SGX enclaves ensures that this sensitive information is processed in a secure, isolated environment, greatly enhancing the security of your password manager.

```
enclave {
    from "Sealing/Sealing.edl" import *;

    trusted {
        /* define ECALLs here. */
        public void ecall_get_account([out] int* result);
        void ecall_create_account();
        void ecall_auth([in, size=master_pass_size] const char* master_pass, size_t master_pass_size);
        public void ecall_verify_master_password([in, size=input_len] const char* input, size_t input_len,
[in, size=actual_len] const char* actual, size_t actual_len, [out] int* result);
    };

    untrusted {
        /* define OCALLs here. */
        void ocall_print([in, string]const char* str);
        void ocall_read_file([out] int* err_code, [in, string] const char* file_path, [out, size=sealed_size]
uint8_t* sealed_data, size_t sealed_size);
        void ocall_get_string([out] int* result, [out, size=buffer_size] char* str_buffer, size_t buffer_size);
        void ocall_remove_file([in, string] const char* file_path);
    };
};
```

Figure 11: Enclave.edl

6 Lessons Learned

Adapting to Advanced Security Technologies - Contextualizing the Learning Curve: The integration of Intel SGX presented a significant learning curve. This experience underlined the complexities inherent in implementing and managing hardware-based security solutions. We delved into the intricacies of SGX technology, ranging from enclave creation to data sealing and unsealing processes.

Security Beyond Encryption - Comprehensive Security Approach: The project highlights that effective security strategies encompass more than just robust encryption algorithms. They also involve critical components like secure storage mechanisms, regular data integrity checks, and the establishment of secure communication channels between different system components.

The Value of Secure Enclave - SGX Enclaves as a Security Paradigm: The utilization of SGX enclaves underscored their potential as a revolutionary approach to data security. These enclaves provided an isolated execution environment, effectively shielding sensitive operations from external threats and system vulnerabilities.

Resilience Against Threats - Building Robust Systems: Our experience reinforced the necessity of developing systems capable of withstanding a diverse range of threats. This encompasses not only software vulnerabilities but also potential hardware-based attacks. The project served as a practical testament to the importance of designing systems with an inherent capacity to resist and recover from various security breaches.

7 Limitation

Hardware Dependency: The reliance of the password manager on Intel SGX technology restricts its usage to only those systems equipped with SGX-enabled processors. This hardware dependency significantly narrows the user base, limiting the application's reach to a specific segment of potential users who possess the necessary hardware capabilities. This limitation not only affects the widespread adoption of the password manager but also restricts its utility in diverse computing environments where such hardware may not be available or feasible to use.

Performance Overheads: The integration of SGX enclaves, while enhancing security, brings with it inherent performance trade-offs. The additional layers of security and the processes of encryption and decryption can introduce latency, affecting the speed and responsiveness of the password manager. Users might experience delays during critical operations like initialization and data retrieval, which can be a deterrent for those requiring fast and seamless access to their credentials.

Complexity and Development Challenges: Developing an application with advanced technologies like SGX is a complex endeavor. It requires specialized knowledge in areas like secure enclave programming and encryption, which may not be readily available. Moreover, the development process can be more time-consuming and costly compared to standard

password management solutions, posing challenges in terms of resource allocation and project timelines.

Scalability Concerns: As the user base grows or the number of stored credentials increases, the password manager may face scalability challenges. Efficiently managing and accessing a large volume of encrypted data within the SGX enclave can become increasingly complex. This scalability issue could impact performance and the overall user experience, especially when dealing with a substantial amount of sensitive data.

Potential SGX Vulnerabilities: Despite the high level of security offered by SGX, it is not immune to vulnerabilities. Past instances have shown that SGX can be susceptible to specific types of attacks, which could potentially compromise the data stored within enclaves. Relying solely on SGX for security puts the system at risk should any new vulnerabilities be discovered in the technology.

Limited Recovery Options: The security protocol of deleting the sealed data file after multiple failed password attempts, while enhancing security, could lead to irreversible data loss if users forget their master passwords. This lack of recovery options might pose a significant challenge for users, particularly those who may not have other means of recovering their stored credentials.

User Experience Challenges: Maintaining a balance between stringent security measures and user-friendly experience is a crucial challenge for the password manager. The necessity for robust security can sometimes lead to complex user interfaces and processes, potentially deterring users who prefer simplicity and ease of use. Ensuring that the application remains accessible and straightforward, without compromising on security, is a critical aspect that needs careful consideration.

Software Compatibility and Integration: The integration of the password manager with a broad spectrum of applications and platforms presents its own set of challenges. Achieving compatibility and seamless integration with various systems, browsers, and devices is a significant development task. This aspect is crucial for ensuring that the password manager can function effectively in diverse technological ecosystems.

Cost Implications: Developing and maintaining a secure system using SGX technology can be costlier than simpler password management solutions. This cost factor could impact the commercial viability of the project, particularly for small-scale users or businesses who may find the investment in such a technology prohibitive. Balancing the cost with the benefits provided by the advanced security features of SGX is essential for ensuring the project's success and accessibility to a wider audience.

8 Future Work

Building on the foundational work of the SGX-based password manager project, several avenues for future development promise to enhance both the security and user experience aspects of the application. One key area of focus is the expansion of cross-platform compatibility. Adapting the password manager to function seamlessly across different operating systems like Windows, macOS, Linux, and mobile platforms would significantly increase its accessibility and utility for a broader range of users. Incorporating multi-factor authentication is another vital upgrade. Adding layers such as biometrics, one-time passwords (OTPs), or hardware tokens would provide an additional security barrier, safeguarding user access even more robustly.

An innovative direction could be the development of an automated password update feature. This functionality would allow the system to regularly update passwords or do so in response to detected security threats, thus maintaining high-security standards without requiring manual input from the user. Cloud integration for secure backup and synchronization represents a significant enhancement, especially in today's interconnected world. Users would benefit from being able to access their credentials across devices, with the assurance that their data remains secure in the cloud. Each of these potential developments not only builds upon the existing strengths of the password manager but also opens up new possibilities for enhancing security and user experience, ensuring the application stays relevant and effective in the face of rapidly evolving digital landscapes.

9 Conclusion

In conclusion, the creation and use of VaultSGX mark a noteworthy advancement in the context of password management considering the constantly changing cybersecurity situation. Through the use of Intel Software Guard Extensions (SGX) technology, this project has effectively created a strong and safe user-centric password management system, which is an environment for managing and storing user credentials. Positioning itself as a forward-thinking solution, VaultSGX not only solves the present problems related to password security but also foresees emerging risks.

The intricate design of VaultSGX incorporates an uncompromising devotion to security. By creating a secure execution environment, the effective use of SGX enclaves strengthens the password management procedure against outside attacks and any vulnerabilities. The project's systematic methodology, which is meticulously described in detail throughout the project phases, ensures the highest security requirements are maintained while guaranteeing a smooth and simple user experience. VaultSGX is more than just a password manager; it's a digital fortress that gives users a secure location for their most private passwords.

References

[1] Hardening password managers with intel ® software guard extensions.

[2] Intel® Software Guard Extensions — intel.com. https://www.intel.com/content/www/us/en/developer/tools/software-guard-extensions/overview.html. [Accessed 15-11-2023].

[3] BREKALO, H., STRACKX, R., AND PIESSENS, F. Mitigating password database breaches with intel sgx. In *Proceedings of the 1st Workshop on System Software for Trusted Execution* (New York, NY, USA, 2016), SysTEX '16, Association for Computing Machinery.

[4] CHAKRABARTI, S., KNAUTH, T., KUVAISKII, D., STEINER, M., AND VIJ, M. Chapter 8 - trusted execution environment with intel sgx. In *Responsible Genomic Data Sharing*, X. Jiang and H. Tang, Eds. Academic Press, 2020, pp. 161–190.

[5] CONDÉ, R. C. R., MAZIERO, C. A., AND WILL, N. C. Using intel sgx to protect authentication credentials in an untrusted operating system. In *2018 IEEE Symposium on Computers and Communications (ISCC)* (2018), pp. 00158–00163.

[6] DA SILVA, M. S. L., DE OLIVEIRA SILVA, F. F., AND BRITO, A. Squad: A secure, simple storage service for sgx-based microservices. In *2019 9th Latin-American Symposium on Dependable Computing (LADC)* (2019), pp. 1–9.

[7] GÖTZFRIED, J., ECKERT, M., SCHINZEL, S., AND MÜLLER, T. Cache attacks on intel sgx. In *Proceedings of the 10th European Workshop on Systems Security* (New York, NY, USA, 2017), EuroSec'17, Association for Computing Machinery.

[8] GUERON, S. A memory encryption engine suitable for general purpose processors. Cryptology ePrint Archive, Paper 2016/204, 2016. https://eprint.iacr.org/2016/204.

[9] KRAWIECKA, K., KURNIKOV, A., PAVERD, A., MANNAN, M., AND ASOKAN, N. Safekeeper: Protecting web passwords using trusted execution environments. In *Proceedings of the 2018 World Wide Web Conference* (Republic and Canton of Geneva, CHE, 2018), WWW '18, International World Wide Web Conferences Steering Committee, p. 349–358.

[10] KRAWIECKA, K., PAVERD, A., AND ASOKAN, N. Protecting password databases using trusted hardware. In *Proceedings of the 1st Workshop on System Software for Trusted Execution* (New York, NY, USA, 2016), SysTEX '16, Association for Computing Machinery.

[11] MCKEEN, F., ALEXANDROVICH, I., ANATI, I., CASPI, D., JOHNSON, S., LESLIE-HURD, R., AND ROZAS, C. Intel® software guard extensions (intel® sgx) support for dynamic memory management inside an enclave. In *Proceedings of the Hardware and Architectural Support for Security and Privacy 2016* (New York, NY, USA, 2016), HASP '16, Association for Computing Machinery.

[12] MOFRAD, S., ZHANG, F., LU, S., AND SHI, W. A comparison study of intel sgx and amd memory encryption technology. In *Proceedings of the 7th International Workshop on Hardware and Architectural Support for Security and Privacy* (New York, NY, USA, 2018), HASP '18, Association for Computing Machinery.

[13] PRIEBE, C., VASWANI, K., AND COSTA, M. Enclavedb: A secure database using sgx. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), pp. 264–278.

[14] RICHTER, L., GÖTZFRIED, J., AND MÜLLER, T. Isolating operating system components with intel sgx. In *Proceedings of the 1st Workshop on System Software for Trusted Execution* (New York, NY, USA, 2016), SysTEX '16, Association for Computing Machinery.

[15] ROSS, B., JACKSON, C., MIYAKE, N., BONEH, D., AND MITCHELL, J. C. Stronger password authentication using browser extensions. In *14th USENIX Security Symposium (USENIX Security 05)* (Baltimore, MD, July 2005), USENIX Association.

[16] SOBCHUK, J., O'MELIA, S., UTIN, D., AND KHAZAN, R. Leveraging intel sgx technology to protect security-sensitive applications. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)* (2018), pp. 1–5.

[17] WANG, C., JAN, S. T. K., HU, H., AND WANG, G. Empirical analysis of password reuse and modification across online service.

[18] WILL, N. C., AND MAZIERO, C. A. Intel software guard extensions applications: A survey. *ACM Comput. Surv. 55*, 14s (jul 2023).

[19] WU, L., CAI, H. J., AND LI, H. Sgx-uam: A secure unified access management scheme with one time passwords via intel sgx. *IEEE Access 9* (2021), 38029–38042.

[20] ZHENG, W., WU, Y., WU, X., FENG, C., SUI, Y., LUO, X., AND ZHOU, Y. A survey of intel sgx and its applications. *Frontiers of Computer Science 15*, 3 (Dec 2020), 153808.