



## **Secure Vault**

**Designed & developed by**

ARELLA MANI KISHOR	2211CS040009
BRAHMARAOUTHU VISHWATEJA	2211CS040024
GORANTLA JAGADEEP	2211CS040055
IMMADI SAI KIRAN	2211CS040059

**GUIDED BY**

**Dr. B. Nageshwar Rao**

Assistant Professor

**Department of Computer Science & Engineering (Cyber Security)**

**II YR - II SEM**

**Malla Reddy University, Hyderabad**

**2022-2026**



# MALLA REDDY UNIVERSITY

(Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)

## **CERTIFICATE**

This is to certify that this is the bonafide record of the application development entitled “**SECURE VAULT**”, submitted by **ARELLA MANI KISHOR (2211CS040009)**, **BRAHMARAOUTHU VISHWATEJA (2211CS040024)**, **GORANTLA JAGADEEP (2211CS040055)**, **IMMADI SAI KIRAN (2211CS040059)** of B. Tech II year II semester, Department of CSE (CS) during the year 2023-24. The results embodied in this report have not been submitted to any other University or institute for the award of any degree or diploma.

**Internal Guide**

**Dr. B. Nageshwar Rao**  
**(Assistant Professor)**

**Head of the Department**

**Dr. G Anand Kumar**  
**CSE (Cyber Security)**

**External Examiner**

## **ACKNOWLEDGEMENT**

We sincerely thank our Head of the Department **Dr.G.Anand Kumar** for his constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the backstage. Last but not the least our sincere appreciation goes to our family who has been tolerant, understanding our moods and extending timely support.

We sincerely express our gratitude towards the mentor **Dr. B.Nageshwar Rao** for guiding us to explore the outcome of our work and we express our sincere gratitude towards them for leading us through the completion of Project.

We wish to express our sincere thanks to **Vice Chancellor sir** and **The Management of Malla Reddy University** for providing excellent infrastructure and their visionary thoughts to prepare ourselves industry ready by focusing on new technologies.

Finally, we would like to thank our family members and friends for their moral support and encouragement to achieve goals.

## ABSTRACT

In today's digital age, the need for robust data protection mechanisms has never been more critical. Secure Vault is an innovative solution designed to address the escalating demands for high-level security in data storage and management. This abstract outlines the key features, underlying technologies, and significant benefits of Secure Vault, providing a comprehensive understanding of its role in safeguarding sensitive information.

Secure Vault employs state-of-the-art encryption algorithms to ensure data is protected both at rest and in transit. By using advanced cryptographic techniques such as AES-256 and RSA-4096, the system guarantees that only authorized users can access the stored data. In addition to encryption, the vault incorporates a multi-layered access control mechanism, including multi-factor authentication (MFA), role-based access control (RBAC), and biometric verification. This ensures that data access is restricted to verified and authorized individuals, reducing the risk of unauthorized access.

Furthermore, Secure Vault maintains comprehensive audit logs that record all access and modification activities. These logs are crucial for monitoring, compliance, and forensic analysis, providing transparency and accountability in data handling. To prevent data loss, Secure Vault includes automated backup solutions and redundancy measures. Data is replicated across multiple locations, ensuring availability even in the event of hardware failure or other disasters.

Overall, Secure Vault represents a cutting-edge solution for secure data storage, combining advanced encryption, stringent access controls, thorough audit trails, and robust redundancy measures. This comprehensive approach not only protects sensitive information but also ensures compliance with regulatory standards and enhances trust in data management practices.

# Index

<b>Contents</b>	<b>Page No.</b>
<b>Chapter 1 Introduction</b> <ul style="list-style-type: none"><li>• 1.1 Problem Definition &amp; Description</li><li>• 1.2 Objectives of the Project</li><li>• 1.3 Scope of the Project</li></ul>	1 - 2
<b>Chapter 2 System Analysis</b> <ul style="list-style-type: none"><li>• 2.1 Background and Literature Survey</li><li>• 2.2 Proposed System</li><li>• 2.3 Software and Hardware Requirements</li><li>• 2.4 Feasibility Study</li></ul>	3 - 5
<b>Chapter 3 Architectural Design</b> <ul style="list-style-type: none"><li>• 3.1 Module Design</li><li>• 3.2 Method and Algorithm Design</li><li>• 3.3 Project Architecture</li></ul>	6 - 12
<b>Chapter 4 Implementation and Testing</b> <ul style="list-style-type: none"><li>• 4.1 Coding Blocks</li><li>• 4.2 Sample Code</li><li>• 4.3 Execution Flow</li><li>• 4.4 Testing</li></ul>	13 - 19
<b>Chapter 5 Results</b> <ul style="list-style-type: none"><li>• 5.1 Resulting Screens</li></ul>	20 - 21
<b>Chapter 6 Conclusions and Future Scope</b> <ul style="list-style-type: none"><li>• 6.1 Conclusion</li><li>• 6.2 Future Scope</li></ul>	22 - 23
<ul style="list-style-type: none"><li>• <b>Bibliography</b></li><li>• <b>Web Link of project</b></li></ul>	24
<ul style="list-style-type: none"><li>• <b>Paper Publications</b></li></ul>	25 - 33

# CHAPTER - 1

## INTRODUCTION

### 1.1 Problem Definition & Description

- **Problem Specification**

- Cyber-attacks on secret vault systems released via Robust Vault continue to grow as more and more vaults are integrated with digital technologies. The vault's software can be vulnerable for hackers to be able to break in. This is typically (but not necessarily) done through methods like phishing (creating user accounts or tricking employees to provide sensitive information) or spreading malware through the system and gives remote control to the attacker.

- **Problem Description**

- But one of the biggest threats to vault security is just as often internal as it is external. Insider threats are about employees or people who have direct access to the vault but take advantage of it for criminal intentions. This could be through direct theft of valuable goods or with external criminals to allow a breach. Vaults can also be threatened with security compromises due to technological failures. Software bugs, hardware malfunctions and system glitches can occur in automated vault systems that are digitally driven.
- In addition to usual pitfalls all vault systems contend with security and functionality issues. They cut across various borders, technological vulnerabilities, human factors, regulatory challenges, environmental problems and financial constraints. Software bugs and hardware malfunctions can very easily ruin the day for automated vault systems which draw large parts of their strength from vulnerable digital infrastructure.

## 1.2 Objectives of the Project

- **Aim of the Project**

- Secure Vault's objective is to protect the confidential data from the unauthorized users and provide access control to the verified user with a private passkey.

- **Tasks and Deliverables**

- Store and encrypt your files in a hidden directory somewhere in your PC.
- The freedom to choose whether to **encrypt** your files before hiding them.
- Your files will be encrypted using AES encryption algorithm while the key being your master password.
- You can even delete and reset the whole vault if you wish to after supplying your master-password.

## 1.3 Scope of the Project

- The secure vault application aims to ensure data security, robust access control, and encryption for sensitive information using Python programming and advanced cryptographic techniques. It restricts data access to authorized users only, implementing strict authentication and authorization processes. All data uploaded to the vault will be encrypted with secure algorithms, making it unreadable without the master key, which users need to enter each time they view their files.
- Effective workflow management will ensure smooth operation, from data upload to access, with user-friendly implementations that make the application easy to use. Users can access their encrypted files anytime using an index value, ensuring both security and ease of use. Additionally, the application will continuously update to adhere to the latest security standards, providing ongoing protection against emerging threats and ensuring data integrity.

## CHAPTER - 2

### SYSTEM ANALYSIS

#### 2.1 Existing System

- **2.1.1 Background and Literature Survey**

- Gokila Dorai, Sudhir Aggarwal, Neet Patel, and Charisa Powell. 2020. VIDE - Vault App Identification and Extraction System for iOS Devices. Forensic Science International: Digital Investigation 33 (Jul 2020), 301007.<https://doi.org/10.1016/j.fsidi.2020.301007>
- Debra Aho Williamson, "Worldwide Social Network Ad Spending: 2011 Outlook", February 2011. [online] <http://www.emarketer.com/Reports/A11/Emarketer-2000-757.aspx>

- **2.1.2 Limitations of Existing System**

- **Limited Compatibility:** Some secure vault applications may have limited compatibility with different operating systems or devices, restricting users' ability to access their data across various platforms.
- **Complexity in Setup and Usage:** Complex setup processes and user interfaces can hinder the adoption of secure vault applications, especially among less tech-savvy users, reducing overall usability and effectiveness.
- **Limited File Type Support:** Some vault applications may have limitations in supporting a wide range of file types or file sizes, restricting users' ability to store and access diverse types of data.



## 2.2 Proposed System

### • 2.2.1 Advantages of Proposed System

- **Encryption:** Encrypted vaults ensure that only authorized people can access your data.
- **Data confidentiality and integrity:** Maintains Data privacy. Stores and manages the user's sensitive data.
- **Shield sensitive data:** vault is a data storage on the computer that can be locked or unlocked with a password.
- **Trust Building:** Maintaining a secure vault with reliability, enhancing trust with customers, partners.

## 2.3 Software and Hardware Requirements

### • 2.3.1 Software Requirements

- Python supported software.
- Windows 11 (32-bit or 64-bit), Windows 10 (32-bit or 64-bit), Windows 8 (32-bit or 64-bit), Windows 7 (32-bit or 64-bit), Windows Server 2008 – 2016.
- Implementation of RSA algorithm.
- Requires master password.

### • 2.3.2 Hardware Requirements

- User experience on laptops or systems.
- User friendly access (authorized access).
- Files are in different locations and are accessed after un-hiding the files from the vault.
- System is required to have a certain storage to save files.
- Easy to operate.

## 2.4 Feasibility Study

- **2.4.1 Technical Feasibility**

- Secure Vault exhibits strong technical feasibility with features like the secure protection and secure master key.
- To enhance access control.
- Secure protection while providing the encryption and protection of data.

- **2.4.2 Robustness and Reliability**

- Offer a variety of protection for valuable data from unauthorized users.
- It is reliable as it is provided with an encryption algorithm and one can control the storage only by a Master key provided by the user.
- Ensure compatibility with various operating systems, devices, and file types to accommodate diverse user needs and environments.
- Employ mechanisms to verify data integrity, such as checksums or cryptographic hashes, to detect and prevent unauthorized modifications or corruption.
- By prioritizing robustness and reliability in the design, implementation, and maintenance of the secure vault application, users can trust that their sensitive data is securely protected and accessible whenever they need it.

- **2.4.3 Economic Feasibility**

- **Open Source Technologies:** Utilize open-source technologies and libraries whenever possible to reduce software development costs and leverage existing solutions for encryption, access control, and data storage.
- **Freemium Model with Upselling:** Offer a free version of the secure vault application with basic features
- **Serverless Architecture:** Consider serverless computing architectures, such as Function as a Service (FaaS), to eliminate the need for managing server infrastructure, reducing operational costs and overhead.

## CHAPTER - 3

### ARCHITECTURAL DESIGN

#### 3.1 Module Design

- **3.1.1 Data Management**

- It is responsible for collecting and storing the data in the vault.
- All data stored in the vault is being encrypted using cryptographic algorithm.
- **End-to-End Encryption:** Implement end-to-end encryption to ensure that data is encrypted from the moment it is uploaded to the vault until it is accessed by authorized users.
- **Strong Encryption Algorithms:** Utilize strong encryption algorithms such as AES (Advanced Encryption Standard) with secure key management practices to protect data at rest and in transit effectively.

- **3.1.2 File Selection Module**

- **File Upload Interface:** Create a user-friendly interface that allows users to easily upload files to the vault.
- **Encryption:** Ensure that files uploaded to the vault are encrypted using strong encryption algorithms before being stored into the vault. Implement end-to-end encryption to protect data both in transit and at rest.
- **Access Control:** Integrate access control mechanisms to restrict access to files based on user authorization.
- **Individual File Selection:** Allow users to select and upload individual files to the vault. Provide options to select files from the local file system or from other cloud storage services.
- **Cross-Platform Compatibility:** Ensure that the file selection module is compatible with various operating systems and devices, allowing users to access and upload files from desktops, laptops, tablets.

- **3.1.3 Design Module**

- Designing a module for a secure vault application involves several key considerations to ensure data security, user privacy, and seamless functionality.
- **User Authentication and Authorization:** Implement a robust authentication system to verify users' identities before granting access to the vault.
- Encrypt files uploaded to the vault using strong encryption algorithms (e.g., AES) to protect data at rest.
- Design a user-friendly interface that allows users to easily navigate the vault and manage their files.
- Provide intuitive file upload, download, and management functionalities.

## **3.2 Method & Algorithm Design**

- **3.2.1 Encryption**

- Encrypting the data stored in the vault ensures that even if unauthorized access is gained, the data remains unreadable. Strong encryption algorithms like AES (Advanced Encryption Standard) are commonly used.
- Before storing data in the vault, it undergoes preparation for encryption. This process may include converting the data into a suitable format, such as binary or plaintext, and segmenting it into fixed-size blocks if necessary.
- **Secure Data Containers:** Store encrypted data within secure containers or vaults protected by strong encryption and access controls.

- **3.2.2 Access Control**

- **Role-Based Access Control (RBAC):** Implement RBAC to define roles and permissions for users accessing the vault. Employ traditional master-key authentication for user login to the vault.

- Implementing strict access control mechanisms ensures that only authorized users can access the vault and only the user with the correct private key can enter and access the vault.
- **Permission Assignment:** Assign specific permissions to each role, dictating what actions users belonging to that role can perform within the vault. Permissions may include read, write, delete, share, and manage permissions.

- **3.2.3 Algorithm (AES (Advanced Encryption Standard)):**

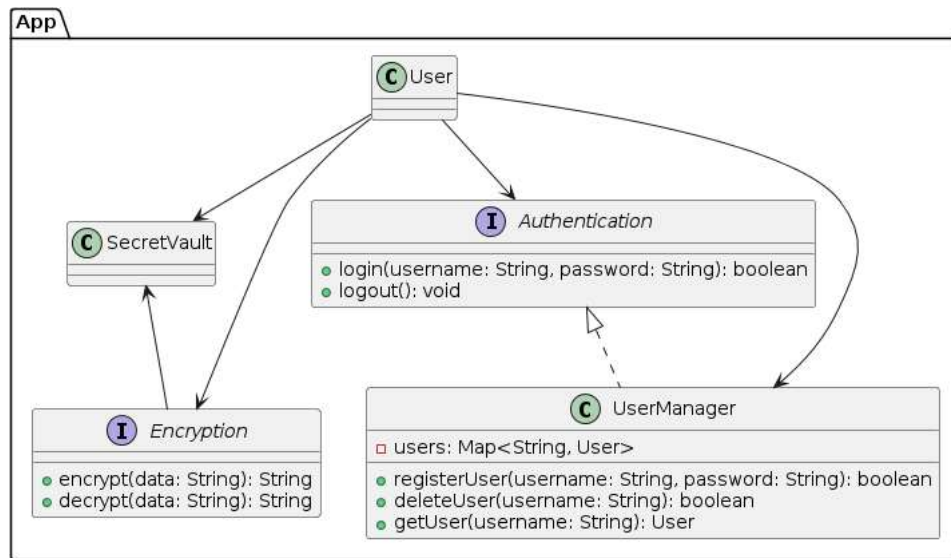
- AES is widely used for encrypting data due to its security, efficiency, and standardization. It stores and provides the security to the files that are added into the vault.
- It is a block cipher that encrypts and decrypts data in fixed-size blocks, typically 128 bits in length, using a symmetric key
- When a user uploads a file to the vault, the data is encrypted using AES encryption before being stored in the vault.
- AES encryption requires a cryptographic key for both encryption and decryption. System vaults typically use a secure key management system to generate, store, and manage encryption keys securely.

- **3.2.4 Secure Storage**

- **Secure Data Containers:** Store encrypted data within secure containers or vaults protected by strong encryption and access controls. Ensure that only authorized users with proper authentication and authorization can access the encrypted data.
- **Checksums and Hashing:** Checksums or cryptographic hash functions are used to verify the integrity of data stored within the vault.
- By implementing these measures, a system vault application can provide secure storage for sensitive data, protecting it from unauthorized access and tampering.

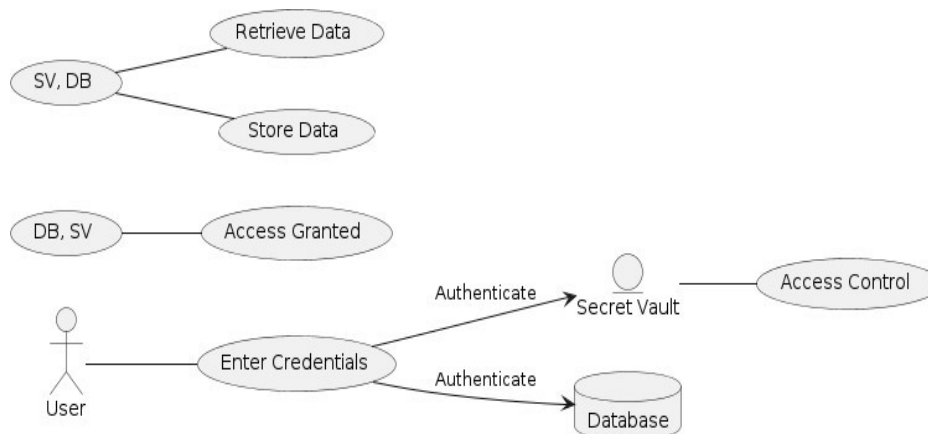
## 3.3 Project Architecture

### • 3.3.1 Architectural Diagram



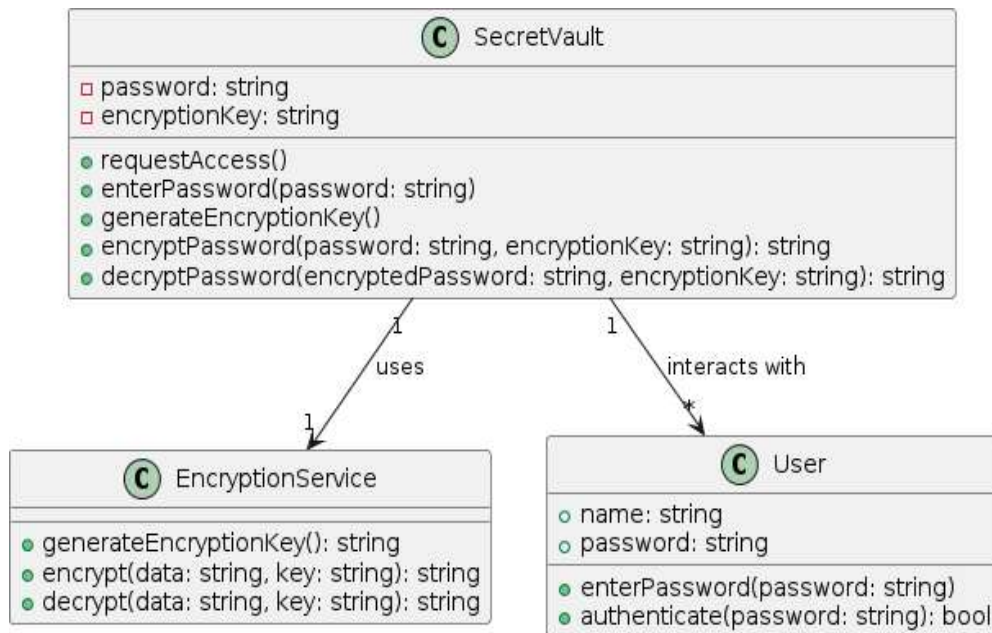
- User has a private key to enter the vault. Only the user can enter the vault with his private key.
- The private key is then authenticated and verifies and validates whether the user is authenticated or not.
- The user manager will provide access if the private key matches with the database.
- The user can encrypt the files before adding them into the vault.

### • 3.3.2 Data Flow Diagram



- The files/data received from the user is allotted in the system space internally.
- The Data is accessed to authorized users only. Storing and retrieving is done safely.
- Retrieval of data is easier and a specific file location is provided.
- Access control is restricted to the other users except for the authenticated user.

### • 3.3.3 Class Diagram

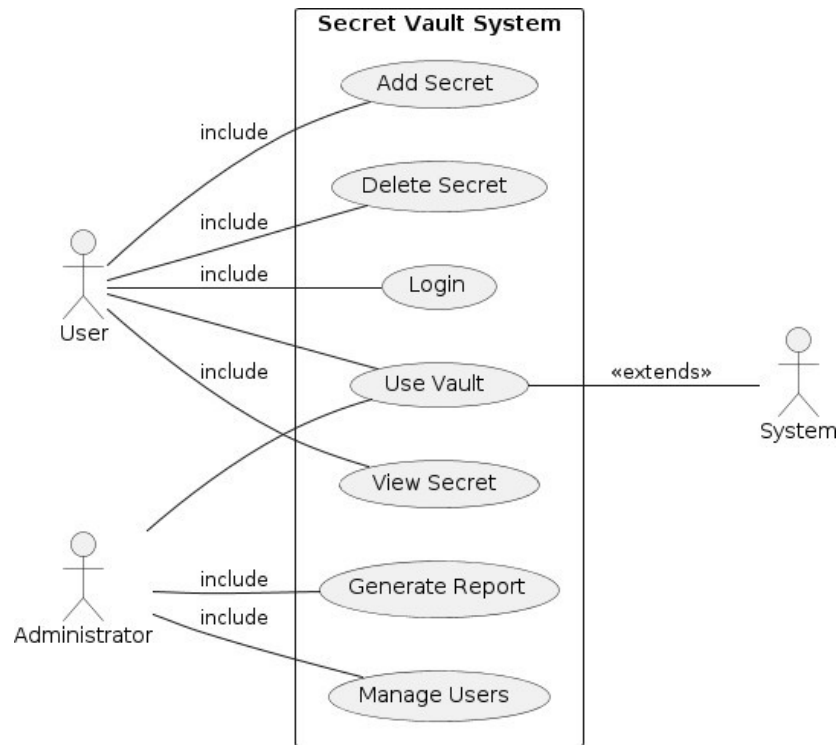


- Secret Vault represents the vault where secrets are stored. It has attributes password and encryption key for managing access and encryption.
- Encryption Service provides encryption and decryption functionality. It has methods generate Encryption Key (), encrypt (), and decrypt ().
- User represents the user of the secret vault application. It has attributes name and password, and methods enter Password () and authenticate ().
- Secret Vault uses Encryption Service for encryption and decryption.

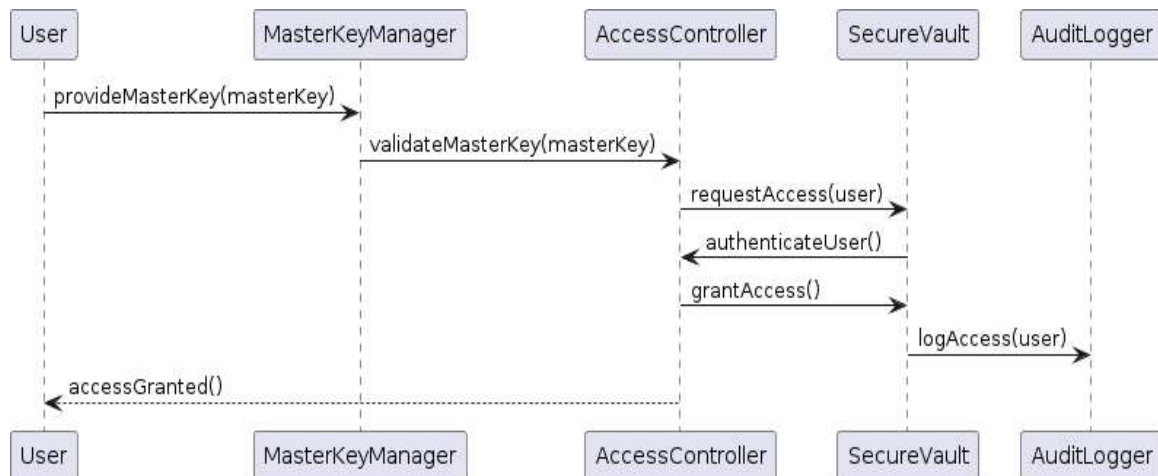
### • 3.3.4 Use Case Diagram

- User: Represents a regular user of the secret vault application.
- Administrator: Represents an administrator who manages the secret vault application.

- Use Cases:
- Login: Both users and administrators need to log in to access the application.
- Add Secret: Users can add secrets to the vault.
- Delete Secret: Users can delete secrets from the vault.
- View Vault: Users can view the contents of the vault.
- Generate Vault: Administrators can generate the vault



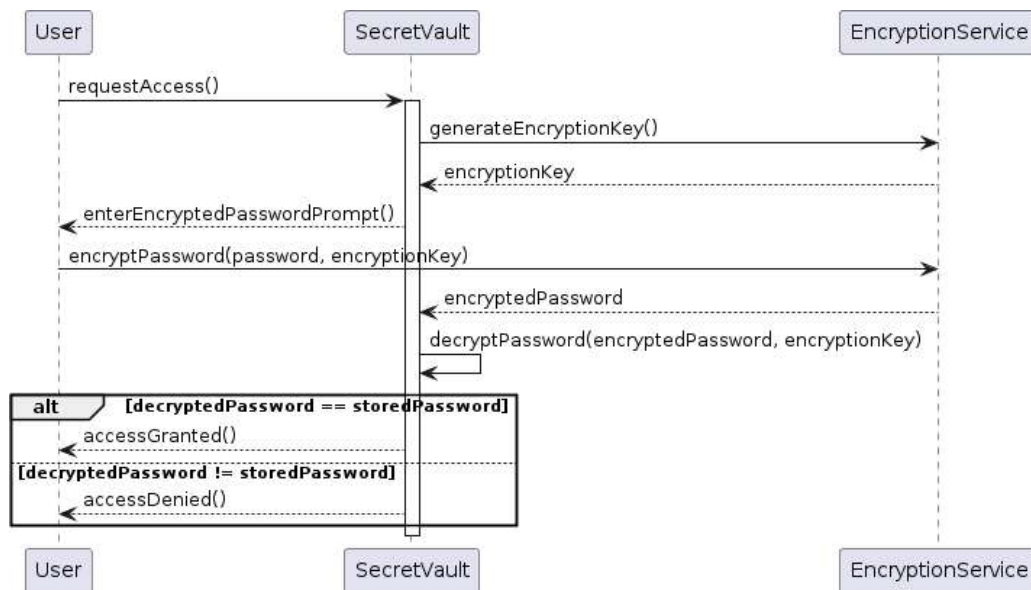
### • 3.3.5 Sequence Diagram





- The User provides the master key to the MasterKeyManager.
- The MasterKeyManager validates the provided master key.
- The Access Controller then requests access to the Secure Vault on behalf of the user.
- The Secure Vault authenticates the user.
- If the authentication is successful, the Access Controller grants access to the Secure Vault.
- Finally, the Secure Vault logs the access event using the Audit Logger and notifies the User that access has been granted.

### • 3.3.6 Activity Diagram



- The process starts at the Start node.
- The application checks if the user exists.
- If the user exists, the application authenticates the user.
- If the authentication is successful, the application displays the main menu.

## CHAPTER - 4

### IMPLEMENTATION AND TESTING

#### 4.1 Coding Blocks

- **Code Block 1**
- Creating class file for secret\_vault

```
class secret_vault:
    buffer_size = 64 * 1024

    def __init__(self, masterpwd):
        self.masterpwd = masterpwd

    def add_file(self, path, encrypt):
        if encrypt:
            filenameWithExt = os.path.basename(path) + '.aes'
            vaultpath = self.hid_dir + filenameWithExt
            pyAesCrypt.encryptFile(path, vaultpath, self.key.decode(),
self.buffer_size)
        else:
            shutil.copy(path, self.hid_dir)

    def del_file(self, index):
        filenameWithExt = self.files[index]
        vaultpath = self.hid_dir + filenameWithExt
        if filenameWithExt.endswith('.aes'):
            filename = filenameWithExt[:-4]
            pyAesCrypt.decryptFile(vaultpath, filename, self.key.decode(),
self.buffer_size)
            os.remove(vaultpath)
        else:
            shutil.copy(vaultpath, filenameWithExt)
            os.remove(vaultpath)

    def list_files(self):
        self.get_files()
        if not self.files:
            print("\nVault is empty!!!")
            return
        maxlen = max([len(x) for x in self.files])
        print("")
        print('-'*(maxlen+10))
        print("index\tfiles")
```

```

print('-'*(maxlen+10))
for i, file in enumerate(self.files):
    print("{}\t|{}".format(i, file))
print('-'*(maxlen+10))

```

- **Code Block 2**

- Generating master\_key using RSA algorithm

```

def generate_key(self,
salt=b"\xb9\x1f}"'S\xa1\x96\xeb\x154\x04\x88\xf3\xdf\x05", length=32):
    password = self.masterpwd.encode()

    kdf = PBKDF2HMAC(algorithm = hashes.SHA256(),
                      length = length,
                      salt = salt,
                      iterations = 100000,
                      backend = default_backend())

    self.key = base64.urlsafe_b64encode(kdf.derive(password))

def get_files(self):
    self.files = os.listdir(self.hid_dir)

def set_hid_dir(self):
    path = '~/vault'
    hid_path = os.path.expanduser(path)
    self.hid_dir = hid_path + '/'

```

## 4.2 Sample Code

- Providing the main() block code with the operations

```

def main():
    print("Welcome to the secret vault!!!")
    path = os.path.expanduser('~/.vaultcfg')
    if os.path.exists(path):
        masterpwd = getpass("Enter your Master Password : ")
        vault = secret_vault(masterpwd)
        vault.generate_key()
        fernet = Fernet(vault.key)
        with open(path, 'rb') as f:
            actual_mpwd = f.read()
        try:

```

```

        fernet.decrypt(actual_mpwd)
        print('Welcome Back')
    except:
        print("Wrong Master Password!")
        exit()
else:
    masterpwd = getpass("Create a Master Password : ")
    vault = secret_vault(masterpwd)
    vault.generate_key()
    fernet = Fernet(vault.key)
    enc_mpwd = fernet.encrypt(masterpwd.encode())
    with open(path, 'wb') as f:
        f.write(enc_mpwd)
        vault.set_hid_dir()
    try:
        os.makedirs(vault.hid_dir[:-1])
    except FileExistsError:
        pass

    if os.name == 'nt':
        call(["attrib", "+H", vault.hid_dir[:-1]])
        call(["attrib", "+H", path])

    print("Welcome")

vault.set_hid_dir()

choice = 0
while choice != 4:
    print("\nEnter 1 to hide a file\nEnter 2 to unhide a file\nEnter 3 to view
hidden files\nEnter 4 to Exit\nEnter 5 to Reset the vault and delete all of its
contents\n")
    try:
        choice = int(input("Enter your choice : "))
    except:
        print("\nUnknown value!")
        continue

    if choice == 1:
        print("\nTip : Drag and Drop the file")
        filepath = input("Enter the path of the file to hide : ")
        filepath = filepath.replace("\\", ")
        if filepath.endswith(' '):
            filepath = filepath[:-1]
        if os.path.exists(filepath):
            if os.path.isfile(filepath):
                while True:
                    enc_or_not = input("Do you want to
encrypt the file? (Y or N) : ")
                    if enc_or_not == 'y' or enc_or_not ==
'Y':
                        print("\nAdding file to the
vault...")

```

```

to the vault")
original file if you want")

'N':
vault...)

to the vault")
original file if you want")

break
elif enc_or_not == 'n' or enc_or_not ==

print("\nAdding file to the

vault.add_file(filepath, 0)
print("\nFile successfully added

print("You can now delete the

break

else:
    print("Type Y or N")

else:
    print("\nGiven path is a directory and not a

file!")

else:
    print("\nFile does not exists!")

elif choice == 2:
    print("")
    try:
        file = int(input("Enter the index of the file from view

hidden files : "))

        vault.del_file(file)
        print("\nFile unhided successfully')
        print('The file will be present in

{}'.format(os.getcwd()))
    except:
        print("\nInvalid index!")

elif choice == 3:
    vault.list_files()

elif choice == 5:
    while True:
        confirm = input("\nDo you really want to delete and

reset the vault?(Y or N) : ")
        if confirm == 'y' or confirm == 'Y':
            pwdCheck = getpass("\nEnter the password to

confirm : ")

            reset = secret_vault(pwdCheck)
            reset.generate_key()
            resetFernet = Fernet(reset.key)
            path = os.path.expanduser('~/.vaultcfg')
            with open(path, 'rb') as f:
                actual_mpwd = f.read()
            try:

```

```

        resetFernet.decrypt(actual_mpwd)
        print('Removing and resetting all
data...')
    except Exception as e:
        print(e)
        print("\nWrong Master
Password!")
        print("Closing program now...")
        exit()
    os.remove(path)
    shutil.rmtree(vault.hid_dir[:-1])
    print("\nReset done. Thank You')
    exit()
    elif confirm == 'n' or confirm == 'N':
        print("\nHappy for that")
        break
    else:
        print("Type Y or N")

if __name__ == '__main__':
    main()

```

## 4.3 Execution Flow

### • 4.3.1 Authentication:

- The user launches the application and is prompted to authenticate themselves, usually by entering a password, PIN, or biometric authentication like fingerprint or facial recognition.
- The application verifies the user's credentials against stored data to grant access.

### • 4.3.2 Main Menu:

- After successful authentication, the user is presented with the main menu of the application.
- The main menu typically provides options for accessing, storing, or managing secret data.

- **4.3.3 Accessing Stored Data:**

- If the user chooses to access stored data, they may be presented with a list of categories or individual items stored in the vault.
- Upon selecting an item, the user may need to provide additional authentication (such as a password) to decrypt the stored data.

- **4.3.4 Storing Data:**

- If the user wants to store new data in the vault, they select an option to add a new item.
- They may be prompted to enter the data they want to store (such as passwords, documents, or other sensitive information).
- The application encrypts the data using strong encryption algorithms before storing it in the vault.

- **4.3.5 Managing Data:**

- The user may have options to edit, delete, or organize the stored data within the application.
- These actions typically require additional authentication to ensure only authorized users can modify the vault contents.

- **4.3.6 Resetting of vault:**

- This new option allows the user to completely erase all data stored within the vault.
- Upon selecting this option, the user may be prompted to confirm their decision to erase all data.
- Once confirmed, the application securely deletes all stored data, ensuring it cannot be recovered.

## 4.4 Testing

- **4.4.1 Test Case 1**

- Calling the vault

```
(.venv) PS C:\Users\User\PycharmProjects\app vault> secret_vault
Welcome to the secret vault!!!
Enter your Master Password :
Enter your Master Password :
```

- **4.4.2 Test Case 2**

- Selecting from the Menu

```
Enter 1 to hide a file
Enter 2 to unhide a file
Enter 3 to view hidden files
Enter 4 to Exit
Enter 5 to Reset the vault and delete all of its contents
Enter 2 to Reset the vault and delete all of its contents
```



## CHAPTER - 5

### Results

#### 5.1 Resulting Screens

- Screen Shot 1

```
(.venv) PS D:\6J\secret_vault> secret_vault
Masterpwd:
Repeat for confirmation:
Welcome to the secret vault!!!
Welcome Back
```

- Screen Shot 2

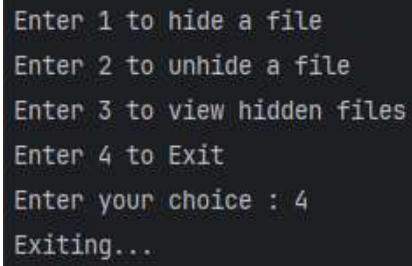
```
Enter 1 to hide a file
Enter 2 to unhide a file
Enter 3 to view hidden files
Enter 4 to Exit
Enter your choice : 3
```

- Screen Shot 3

```
Enter your choice : 3

-----
index    |files
-----
0        |Family Man.mkv.aes
-----
```

- **Screen Shot 4**



```
Enter 1 to hide a file
Enter 2 to unhide a file
Enter 3 to view hidden files
Enter 4 to Exit
Enter your choice : 4
Exiting...
```

## **CHAPTER - 6**

### **Conclusions and Future Scope**

#### **6.1 Conclusion**

- System vaults are used in many different fields, such as the safe keeping of cryptographic keys used in applications to encrypt and decrypt private information. Like a bank vault, a secure is a special secure. It is a virtual or maybe than a genuine secure where you can store records that require to be kept mystery from prying eyes. It was actualized to ensure touchy records and reports from prying eyes. Each record has an area and is continuously accessible. Each client is given a special key to get to the store where they can store all sorts of records counting PPTs, HTML web pages, records, pictures (JPG, PNG, GIF), APK records and more.
- The term "system vault" usually refers to happen within an operating system or software program where private data-such as passwords, cryptographic keys, or other credentials is encrypted and safeguarded. Protecting this sensitive data from theft, alteration, or unauthorized access is the goal of a system vault.
- Thus, in addition to representing an incident in the continuity of advancement in technology and invention, the secure vault application represents the absolute attainment of relevance to security within today's universe.

## 6.2 Future Scope

- Firstly, the application can explore integration with emerging technologies such as blockchain for immutable record-keeping and decentralized authentication protocols. This integration would enhance the security and transparency of the vault, ensuring that user assets remain protected against even the most sophisticated threats.
- Secondly, there is potential for the application to incorporate enhanced collaboration features, allowing users to securely share sensitive information and assets with trusted individuals or teams. This would facilitate seamless collaboration while maintaining robust security measures to prevent unauthorized access.
- Thirdly, leveraging artificial intelligence and machine learning algorithms, the application can develop advanced threat detection capabilities. By continuously analysing user behaviour and monitoring for suspicious activities, the vault can proactively identify and mitigate potential security breaches in real-time.
- Furthermore, ensuring cross-platform compatibility across various devices and operating systems would extend the reach of the application, enabling users to access their vault securely from anywhere, anytime.

# BIBLIOGRAPHY

## References

- **Ferguson, N., Schneier, B., & Kohno, T. (2010).** *Cryptography Engineering: Design Principles and Practical Applications*. Wiley.
- **Gollmann, D. (2011).** *Computer Security*. Wiley.
- **Kaufman, C., Perlman, R., & Speciner, M. (2002).** *Network Security: Private Communication in a Public World*. Prentice Hall.

## Papers

- **Kelsey, J., & Kohno, T. (2006).** *Herding Hash Functions and the Nostradamus Attack*. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2006, 183-200.
- **Microsoft.** *Secure Coding Guidelines*. Retrieved from <https://docs.microsoft.com/en-us/security/develop/>
- **Google. (2021).** *Android Security Overview*. Retrieved from <https://source.android.com/security/>
- **Smith, J. A. (2020).** *Design and Implementation of Secure Vault Systems for Mobile Devices*. Master's thesis, University of California, Berkeley.

## WEB LINK

- [https://github.com/jagadeep18/Secure\\_vault](https://github.com/jagadeep18/Secure_vault)
- <http://127.0.0.2:8080/>



## **A Comprehensive Research into the Role of Secure Vault in Safeguarding Digital Assets.**

<sup>1</sup>A. MANI KISHOR, <sup>2</sup>B. VISHWA TEJA, <sup>3</sup>G. JAGADEEP, <sup>4</sup>I. SAI KIRAN

<sup>1,2,3,4</sup> Student, Department of Computer Science and Engineering-Cybersecurity, Malla Reddy University, Hyderabad, Telangana, India

<sup>1</sup>2211cs040009@mallareddyuniversity.ac.in, <sup>2</sup>2211cs040024@mallareddyuniversity.ac.in,

<sup>3</sup>2211cs040055@mallareddyuniversity.ac.in, <sup>4</sup>2211cs040059@mallareddyuniversity.ac.in

<sup>5</sup>Associate professor, Department of Computer Science and Engineering-Cybersecurity, Malla Reddy University, Hyderabad, Telangana, India

<sup>5</sup>nagesh.south@gmail.com

### **ABSTRACT**

Like a bank vault, a secure is a special secure. It is a virtual or maybe than a genuine secure where you can store records that require to be kept mystery from prying eyes. It was actualized to ensure touchy records and reports from prying eyes. They are put away in the framework and ensured by a secret word known as it was to the client. Since record security is imperative, we scramble the information utilizing encryption procedures and keep it in capacity. All frameworks can utilize this store and all sorts of records can be secured notwithstanding of record estimate limits. Each record has a area and is continuously accessible. Each client is given a special key to get to the store where they can store all sorts of records counting PPTs, HTML web pages, records, pictures (JPG, PNG, GIF), APK records and more.

**Key words:** Framework, scramble, encryption, record, vault, client

### **1. INTRODUCTION**

A secure vault is a scrambled put, where all your vital qualifications are securely put away. All data that you include there is scrambled at your gadget level some time recently it comes to our servers. Indeed we, as a benefit supplier, can't get to it, meaning it's secure from snoopers. It is aiming to ensure their substance from robbery, unauthorized utilize, fire, characteristic fiascos, and other dangers, much like a secure. Not at all like safes, vaults are an indispensably portion of the building inside which they are built, utilizing armored dividers and a firmly designed entryway closed with a complex bolt. Vaults work by scrambling each mystery to help avoid unauthorized clients from picking up get to. They work for the most part as a dynamic capacity holder for insider facts as well as an account administration framework for managing with different advantaged accounts over the company. Vault is an identity-based insider facts and encryption administration framework. A mystery is anything that you need to firmly

control get to such as API encryption keys, passwords, and certificates. Vault gives encryption administrations that are gated by confirmation and authorization strategies. The essential reason of a secure vault is to give an exceedingly secure environment for putting away important things such as cash, adornments, imperative reports, delicate information, valuable metals, and other high-value resources. Vaults are built to withstand a wide run of dangers, common calamities and indeed fear monger attacks. Vaults are built utilizing vigorous materials such as strengthened steel, concrete, and other high-strength amalgams. The dividers, floors, and ceilings are planned to stand up to constrained passage and altering. Get to the vault is firmly controlled and limited to authorized staff as it were. Get to logs and review trails may be kept up to track who enters the vault and when. Vaults are frequently planned to be measured and versatile, permitting for customization and development based on the security needs of the organization.

### 1.1 Introduction to Vault

Vaults are for more than reasonable essential usernames and passwords. Endeavors all over utilize a wide grouping of affirmation rebellious, checking tokens, SSH keys, and certificates to title a few. We imply to all these as “secrets.” Vaults work by scrambling each riddle to offer help expect unauthorized clients from picking up get to. They work for the most part as an energetic capacity holder for insider truths as well as an account organization system for overseeing with distinctive advantaged accounts over the company. A vault is suggested to be a centralized put to manage account assents and favored experiences.

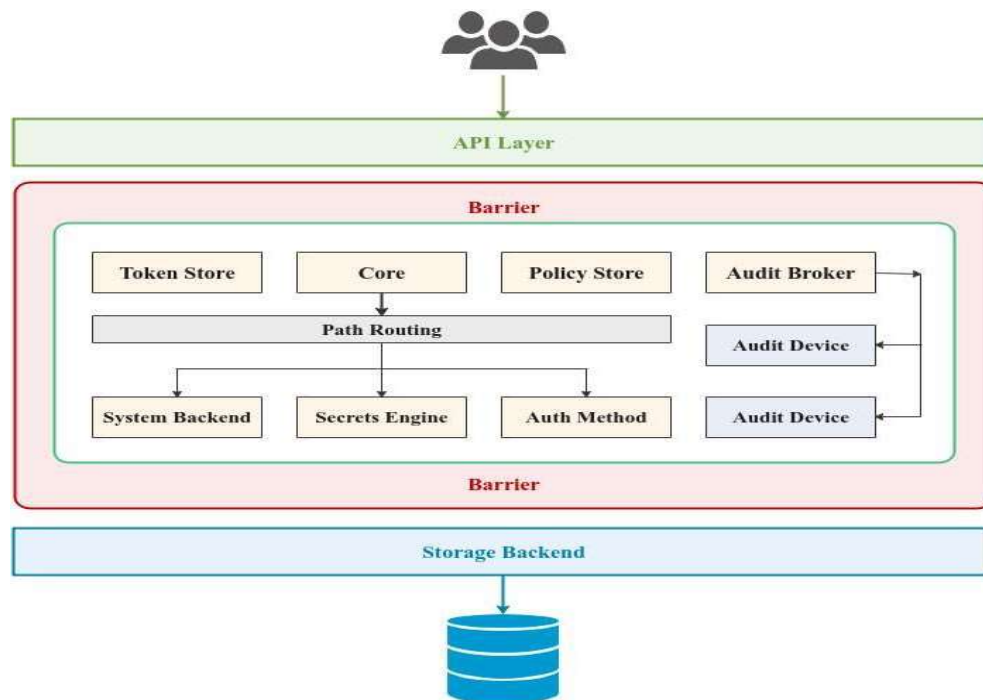
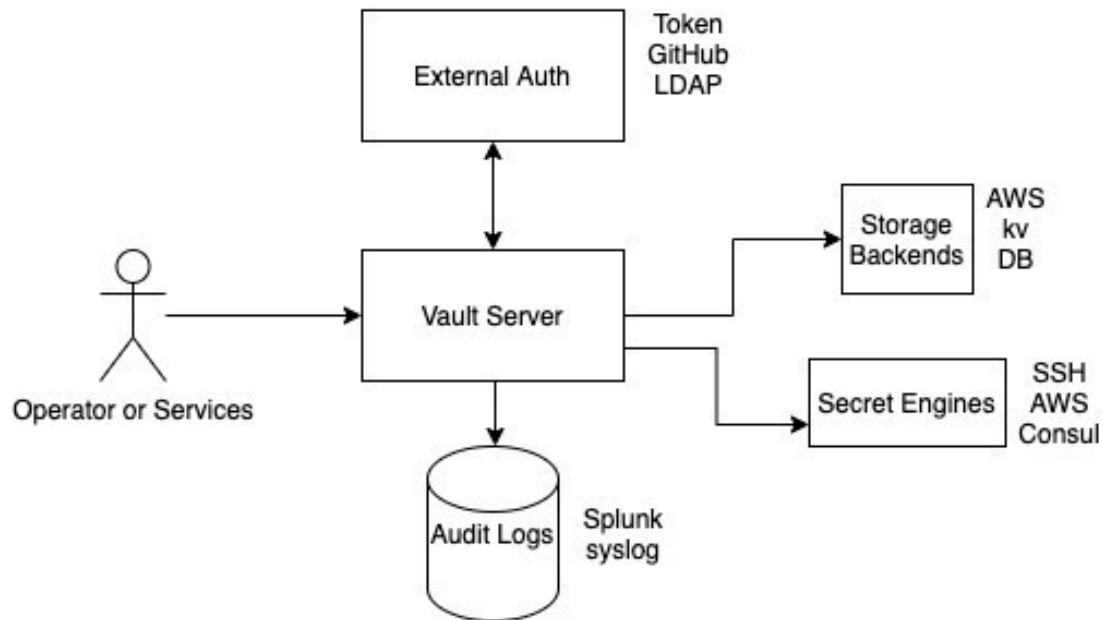


Fig:1 Architecture

### 1.1.1 The Basics of Vault

A vault, in the setting of computer security and encryption, is a framework that stores and oversees insider facts, which are touchy information such as passwords, encryption keys, and tokens. Vaults give a bound together interface to these privileged insights whereas guaranteeing secure capacity, tight get to control, and nitty gritty review logs. They can moreover give energetic mystery era, information encryption, and repudiation capabilities, making it less demanding for organizations to oversee their privileged insights and ensure their information.



**Fig:2 Working of the Vault**

## 2. LITERATURE SURVEY

In computer security, the concept of a framework vault has risen as a pivotal component for shielding delicate information inside working frameworks. A framework vault, basically, serves as a secure capacity framework outlined to ensure basic data such as passwords, cryptographic keys, and qualifications from unauthorized get to or altering. Understanding the standards and usage of framework vaults is basic in guaranteeing the astuteness and secrecy of information inside computing situations. At its center, a framework vault works based on a few essential standards of secure capacity plan. These standards incorporate solid encryption calculations, get to control components, secure key administration, and tamper-resistant capacity. By following to these standards, framework vaults guarantee that information put away inside them remains secured against different dangers and vulnerabilities. A few challenges stand up to the plan and sending of framework vaults, counting versatility concerns, compatibility issues over different stages, and advancing risk scenes. Tending to these challenges requires continuous investigate and advancement, with a center on progressing the convenience, flexibility, and interoperability of framework vaults. Future headings may include investigating novel encryption procedures, improving key administration hones, and joining framework vaults with developing innovations such as blockchain and decentralized character frameworks.

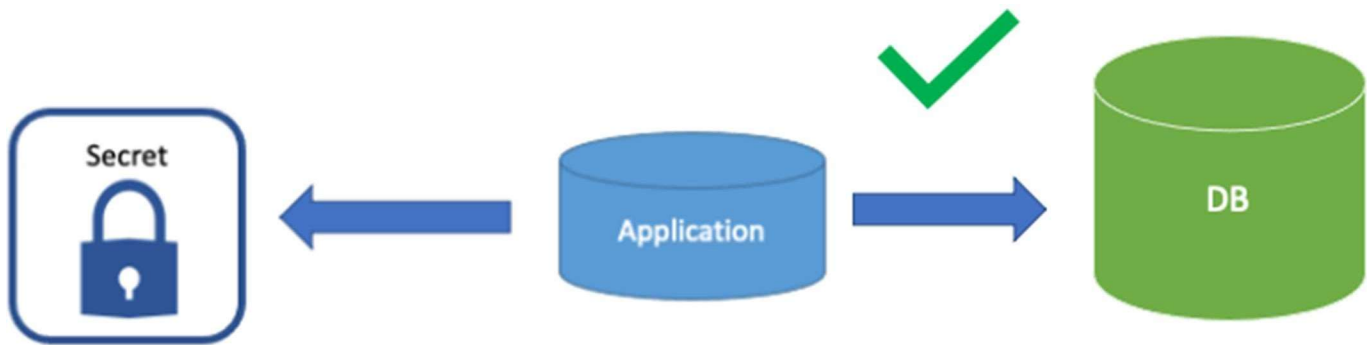


### 3. PROPOSED SYSTEM

Our solution is to keep up integrity and secrecy as information is vital for anybody. We execute a few cryptographic calculations and guarantee that the information is secured and is made accessible to the client at any time. We give security to the records independent of their estimate and keep up in uncovered area inside the framework. The records after retrieval can be found at an area which is seen by the authorized client. Since there may be secrets in the files, access control is restricted to a specific user who has been granted access to the vault. The data is encrypted. After being encrypted, the files are kept secret inside a specific location within the system that is unknown to any user, not even those with permission. Once the files have been decrypted, the user can view their location, send, modify, and then secure them once more. The internal memory will occupy this vault's storage space. Data loss won't occur because access to the vault cannot be intercepted. Since the private key serves as the vault's door, accessing it requires having one. If the user is approved, the vault opens, allowing them to view, restore, and alter all the files within. First in, first out (FIFO) order governs how the files are kept in the vault. When a new file is added, its serial number starts at index 0 and ends at file N.

### 4. WORKING OF VAULT

When discussing computers and security, the term "system vault" usually refers to happen within an operating system or software programme where private data—such as passwords, cryptographic keys, or other credentials is encrypted and safeguarded. Protecting this sensitive data from theft, alteration, or unauthorized access is the goal of a system vault. Whether digital or real, a vault serves as a safe place to store goods that is intended to keep valuables, records, information, or assets safe from theft, damage, loss, or unwanted access. Thick steel, concrete, or reinforced alloys are used in the physical construction of vaults to provide them with resistance against forced entrance, drilling, cutting, and other types of tampering. Robust encryption techniques and access control measures are the foundation of digital vaults' data protection. Only those who have been granted authorization can enter the vault. Access control mechanisms guarantee that the vault's contents can only be accessed by authorized personnel. Within the operating system or software programme, the sensitive data is kept in a protected area. This storage space is intended to prevent unwanted access and is frequently segregated from other system components. Strong encryption algorithms are used to encrypt the data before it is kept in the vault. This guarantees that an attacker will be unable to decode the data without the decryption key, even if they get to access the stored data. The protection of the System Vault also requires careful consideration of physical security issues. This could involve taking precautions against environmental threats, keeping an eye out for physical tampering, and implementing access controls to the actual hardware housing the vault. Frequent backups guarantee that data may be recovered from a backup copy if the primary vault is corrupted or rendered unusable. For auditing purposes, System Vault-related activities like updates, errors, and access attempts are frequently logged.



**Fig 3: Pathway**

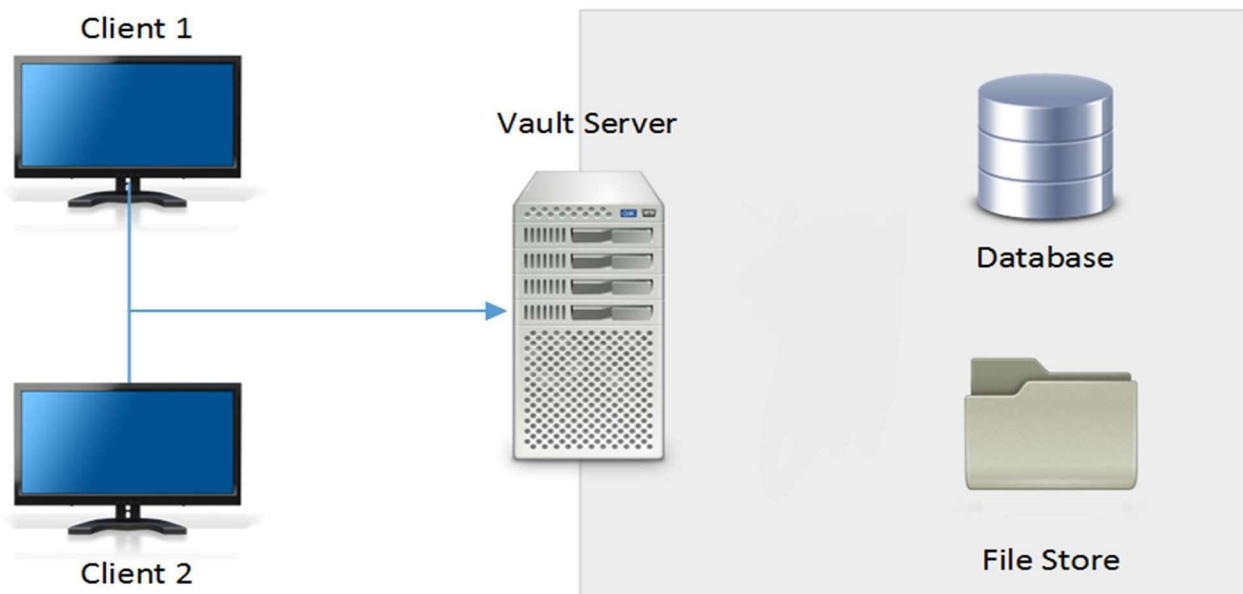
## 5. SOFTWARE & HARDWARE TOOLS USED

### 5.1 Hardware Requirements:

- CPU: 1.6 GHz or higher (minimum), 3 GHz or higher
- Memory: 4 GB RAM (minimum), 8 GB RAM
- Disk space: 10 GB disk space (minimum), 30 GB disk space

### 5.2 Software Requirements:

- Python supported system
- Windows 11 (32-bit or 64-bit), Windows 10 (32-bit or 64-bit), Windows 8 (32-bit or 64-bit), Windows 7 (32-bit or 64-bit), Windows Server 2008 – 2016



**Fig 4: User Interaction**

## 6. RESULTS & DISCUSSION

Secure storage for private data, including passwords, access tokens, and cryptographic keys, is provided by a system vault. Its main goal is to keep sensitive data safe from theft, alteration, and unauthorized access. To protect the stored data, system vault designs usually use strong encryption methods and access controls. Redundancy and failover measures could be incorporated to provide high availability and data integrity.

```
(.venv) PS C:\Users\User\PycharmProjects\app vault> secret_vault
Welcome to the secret vault!!!
Enter your Master Password :
```

**Fig 5: Calling the vault**

```
Enter 1 to hide a file
Enter 2 to unhide a file
Enter 3 to view hidden files
Enter 4 to Exit
Enter 5 to Reset the vault and delete all of its contents
```

**Fig 6: Choices**

```
Enter your choice : 1

Tip : Drag and Drop the file
Enter the path of the file to hide : C:/Users/User/Desktop/GJ/test.txt
Do you want to encrypt the file? (Y or N) : y
```

```
Adding file to the vault...

File successfully added to the vault
You can now delete the original file if you want
```

**Fig 7: Adding a file to Vault**

```
Enter your choice : 3

-----
index    |files
-----
0        |profiledetails.pdf.aes
-----
1        |test.txt.aes
-----
2        |test2.txt.aes
-----
3        |test3.txt.aes
-----
```

**Fig 8: Viewing hidden files in Vault**

```
Enter your choice : 2

Enter the index of the file from view hidden files : 1

File unhided successfully
The file will be present in C:\Users\User\PycharmProjects\secret_vault
```

**Fig 9: Unhiding a File from Vault**

```
Enter your choice : 5

Do you really want to delete and reset the vault?(Y or N) : y

Enter the password to confirm :
Removing and resetting all data...

Reset done. Thank You
```

**Fig 10: Resetting Vault**

System vaults are used in many different fields, such as the safe keeping of cryptographic keys used in applications to encrypt and decrypt private information. Maintaining the integrity of the vault while allowing authorized workers access is contingent upon striking a balance between security and usability. Patches and upgrades on a regular basis are necessary to handle new security threats and vulnerabilities. Workflow interruptions should be prevented by seamless integration with current systems and apps. Disaster recovery and backup plans must be implemented to reduce the possibility of data loss because of cyberattacks, natural catastrophes, or hardware malfunctions.

## **7. CONCLUSION**

The system vault, which acts as a stronghold for protecting sensitive data, is a fundamental component of contemporary cybersecurity infrastructure. Its strong architecture, which includes access restrictions, security safeguards, and cutting-edge encryption techniques, guarantees the availability, confidentiality, and integrity of vital data. The system vault is essential for compliance adherence, risk minimization, and data protection in a wide range of applications spanning industries and sectors, from finance to healthcare and beyond. Nonetheless, difficulties still exist, necessitating a careful balancing act between security and accessibility in addition to constant adaptability to threats that change over time. In an increasingly interconnected world, the system vault is still a vital tool for strengthening defenses and maintaining trust as organizations traverse the complicated terrain of digital security.

## **8. FUTURE SCOPE**

In the future, as technology advances and security requirements change, the capabilities of a "System Vault" or other comparable secure storage techniques will probably change as well. The encryption techniques employed in System Vaults may grow increasingly complex and reliable as processing power rises. To combat new threats, this can entail implementing innovative cryptographic approaches or quantum-resistant algorithms. In addition to conventional passwords or cryptographic keys, biometric authentication techniques like fingerprint or facial recognition may become more and more important in future System Vaults. Access controls could be strengthened even more with multi-factor authentication. System Vaults are expected to cover developments in encryption, authentication, distributed computing, artificial intelligence (AI)-driven security, and regulatory compliance in the future. These developments are intended to tackle new security threats and changing security issues in a world that is becoming more digitally linked and interconnected.

## **9. REFERENCES**

- [1] Gokila Dorai, Sudhir Aggarwal, Neet Patel, and Charisa Powell. 2020. VIDE - Vault App Identification and Extraction System for iOS Devices. Forensic Science International: Digital Investigation 33 (Jul 2020), 301007. <https://doi.org/10.1016/j.fsidi.2020.301007>
- [2] Debra Aho Williamson, "Worldwide Social Network Ad Spending: 2011 Outlook", February 2011. [online] <http://www.emarketer.com/Reports/A11/Emarketer-2000-757.aspx>

- [3] S. S. and P. Srinivasan, "Internet of Things Based Digital Lock System," J. Comput. Theor. Nanosci., vol. 15, 2018.
- [4] J. Jeong, "A Study on the IoT Based Smart Door Lock System," Inf. Sci. Appl., 2016.
- [5] "Security and Usability Improvement on a Digital Door Lock System based on Internet of Things," Int. J. Secur. Its Appl., vol. 9, 2015.
- [6] D. Levin and S. Arafah, "THE DIGITAL DISCONNECT THE WIDENING GAP BETWEEN INTERNET-Prepared by," J. Card. Fail., 2008.
- [7] S. Firdosh, P. Kashyap, B. Durgam, N. Begum, and S. Kumar Singh, "Password Based Door Locking System Using Microcontroller," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. © 2017 IJSRCSEIT, vol. 3, no. 10, pp. 428–432, 2017.
- [8] <https://developer.hashicorp.com/vault/docs/what-is-vault>