Cipher Guard

Protecting the user password keys at rest (on the Disk)

Protecting the user password keys at rest (on the Disk)

PROBLEM STATEMENT 15 :

A. Sanjai

Introduction :

Scope : Developing an application for file encryption which is in turn protected by user pass phrase

Description: Develop an authorization application which in turn protects the password keys. Following are the high level feature

1. Encrypt [AES-256] a user chosen file or directory using a random key a.k.a File Encryption Key
2. Store the random key in a file which has to be protected via user pass phrase.
3. The user pass phrase as well as the random key can not be stored in plain form in the text file.
4. If the user pass phrase authentication is successful retrieve i.e decrypt the file using File Encryption Key.

Hints : You can use user pass phrase as a seed to generate deterministic keys using standard KDF (Key Derivation Function)

original concepts Breakdown (solution) :

1 . Multiple FEK Management:

CIPHERGUARD ALLOWS USERS TO CREATE AND MANAGE MULTIPLE FEKS, EACH ASSOCIATED WITH SPECIFIC FILES/FOLDERS.

2 . User-Friendly GUI (Graphical User Interface):

A CLEAN AND INTUITIVE INTERFACE MAKES ENCRYPTION EASY FOR EVERYONE.

GUI FEATURES: INTUITIVE GUI WITH FEATURES LIKE CANCEL BUTTON, MULTIPLE SELECTION OPTIONS, AND PROGRESS INDICATION.

3 . Robust Error Handling:

GRACEFULLY HANDLES ERRORS DURING ENCRYPTION AND DECRYPTION.

PROVIDES INFORMATIVE ERROR MESSAGES TO GUIDE THE USER.

4 . Securely Store Sensitive Notes with Password Protection:

PASSWORDS, PERSONAL DATA, OR OTHER CONFIDENTIAL NOTES CAN BE SECURED WITHIN THE APPLICATION.

5 . Secure History Logging for Accountability and Security:

CIPHERGUARD MAINTAINS AN ENCRYPTED HISTORY OF ENCRYPTION AND DECRYPTION OPERATIONS.

Encryption Algorithms

THE ENCRYPTION APPLICATION SUPPORTS TWO ADVANCED ENCRYPTION SCHEMES, ENSURING ROBUST SECURITY FOR YOUR FILES AND DIRECTORIES.

ChaCha20-Poly1305 with Argon2id

ChaCha20-Poly1305:

CHACHA20-POLY1305 IS A HIGH-SPEED, AUTHENTICATED ENCRYPTION ALGORITHM THAT COMBINES THE CHACHA20 STREAM CIPHER WITH THE POLY1305 MESSAGE AUTHENTICATION CODE (MAC).

Argon2id:

ARGON2 IS A MEMORY-HARD KEY DERIVATION FUNCTION DESIGNED TO RESIST BRUTE-FORCE ATTACKS, INCLUDING THOSE USING SPECIALIZED HARDWARE SUCH AS GPUS AND ASICS.

AES-256-GCM with scrypt

AES-256-GCM:

ADVANCED ENCRYPTION STANDARD (AES) WITH A 256-BIT KEY IN GALOIS/COUNTER MODE (GCM). AES-256-GCM PROVIDES BOTH ENCRYPTION AND AUTHENTICATION

scrypt:

SCRYPT IS A KEY DERIVATION FUNCTION DESIGNED TO BE MEMORY-INTENSIVE TO THWART LARGE-SCALE CUSTOM HARDWARE ATTACKS. IT REQUIRES SIGNIFICANT MEMORY RESOURCES, MAKING IT DIFFICULT FOR ATTACKERS TO PERFORM PARALLEL BRUTE-FORCE ATTACKS.

Features offered :

1 . File and Folder Encryption/Decryption:

ENCRYPT AND DECRYPT INDIVIDUAL FILES OR ENTIRE DIRECTORIES.

LARGE FILE ENCRYPTION (UP TO 100 GB) HANDLES LARGE FILES AND FOLDERS WITH EASE AND EFFICIENCY.

2 . Encryption Algorithms:

SUPPORTS CHACHA20-POLY1305 WITH ARGON2ID AND AES-256-GCM WITH SCRYPT.

3 . Multiple Attempts Tracking:

- TRACKS INCORRECT PASSPHRASE ATTEMPTS AND LOCKS THE USER OUT AFTER MULTIPLE FAILED ATTEMPTS.

4 . Robust Error Handling:

GRACEFULLY HANDLES ERRORS DURING ENCRYPTION AND DECRYPTION.

5 . Strong Passphrase Validation:

HELPS YOU CREATE STRONG PASSPHRASES USING ZXCVBN FOR ADDITIONAL PASSWORD STRENGTH ANALYSIS.

6 . Note Encryption:

CIPHERGUARD ALLOWS USERS TO SECURELY STORE SENSITIVE NOTES USING A SEPARATE PASSWORD. .

7 .Secure History Logging :

CIPHERGUARD MAINTAINS AN ENCRYPTED HISTORY OF ENCRYPTION AND DECRYPTION OPERATIONS.

ALLOWS USERS TO CLEAR SELECTED HISTORY ENTRIES THROUGH A DIALOG WITH CHECKBOXES..

Process flow :

I. User Interaction:

1 . Launch the Application : The user starts CipherGuard.

2 . Login: THE USER IS PRESENTED WITH A LOGIN PAGE. THEY ENTER THEIR USERNAME AND PASSWORD.

3 . Enter Passphrase: THE USER ENTERS A PASSPHRASE INTO THE PASSPHRASE INPUT FIELD. THE APPLICATION PROVIDES A STRENGTH INDICATOR TO GUIDE THEM.

4 . Choose Encryption Algorithm: THE USER SELECTS EITHER AES-256-GCM OR CHACHA20-POLY1305.

5 . Select Files/Folders: THE USER CHOOSES FILES OR FOLDERS USING THE FILE SELECTION BUTTONS OR DRAG-AND-DROP.

6. Initiate Encryption/Decryption: THE USER CLICKS EITHER THE "ENCRYPT" OR "DECRYPT" BUTTON.

II. Internal Processes

1 . Key Derivation: THE APPLICATION USES THE SELECTED KDF (SCRYPT OR ARGON2ID) AND THE USER'S PASSPHRASE TO DERIVE A STRONG KEY.

2 . FEK Management:

Encryption: If encrypting: A NEW, RANDOM 32-BYTE FEK IS GENERATED. THE FEK IS ENCRYPTED USING THE DERIVED KEY, THE SELECTED ENCRYPTION ALGORITHM, AND A SALT.

Decryption: If decrypting: THE FEK IS DECRYPTED USING THE DERIVED KEY AND THE STORED ALGORITHM. THE ENCRYPTED FEK, SALT, AND ALGORITHM FOR THE SELECTED FILE/FOLDER ARE RETRIEVED FROM "FEK_DATA.BIN".

Technologies Used:

1. Python 3.10+ :
2. PyQt5: FOR THE GRAPHICAL USER INTERFACE.
3. cryptography: FOR IMPLEMENTING ENCRYPTION AND DECRYPTION.
4. argon2-cffi : FOR KEY DERIVATION USING ARGON2ID.
5. scrypt : FOR KEY DERIVATION USING SCRYPT.

Installation

```
pip install PyQt5 cryptography argon2-cffi scrypt
```

conclusion :

THIS FILE ENCRYPTION APPLICATION LEVERAGES ADVANCED ENCRYPTION ALGORITHMS AND KEY DERIVATION FUNCTIONS TO PROVIDE ROBUST SECURITY FOR SENSITIVE AND REGULAR DATA.

BY INCORPORATING BOTH CHACHA20-POLY1305 WITH ARGON2ID AND AES-256-GCM WITH SCRYPT, THE APPLICATION OFFERS USERS A POWERFUL TOOL FOR PROTECTING THEIR FILES AND DIRECTORIES.

Cipher Guard

THANK YOU !!