Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Spring, Year: 2023), B.Sc. in CSE (Day)

File Encryption Decryption Using Shell Script

Course Title: Operating System Lab
Course Code: CSE - 310
Section: 203D1

Students Details

Name	ID
Mohammad Rabiul Hasan	203002024

Submission Date: 21 - 06 - 2023
Course Teacher's Name: Md. Jahidul Islam

[For teachers use only: Don't write anything inside this box]

Lab Project Status

Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

The project "Shell script for file encryption" aims to create a script using Shell programming to encrypt files for enhanced security and privacy. It involves accepting user input for the file to be encrypted and the encryption key, checking file existence, generating encryption keys if necessary, using cryptographic algorithms to encrypt the file's contents, securely handling the encryption key, and providing feedback to the user. The project highlights the use of Shell scripting to automate file encryption tasks and strengthen data protection.

1.2 Motivation

- **Enhanced Security:** File encryption adds an extra layer of security by transforming the contents of a file into an unreadable format. This ensures that even if the file is compromised or intercepted, the data remains inaccessible without the decryption key.
- **Privacy Protection:** Encryption helps safeguard personal and sensitive information, such as financial records, medical data, or confidential business documents. By encrypting files, individuals and organizations can prevent unauthorized individuals or malicious actors from accessing and exploiting their private data.

1.3 Problem Definition

1.3.1 Problem Statement

The problem at hand is the need to secure sensitive files and data from unauthorized access and interception. Traditional file storage and transmission methods may not provide adequate protection, leaving the information vulnerable to potential breaches. There is a requirement for a practical and automated solution that enables users to encrypt their files easily and effectively, ensuring the confidentiality and integrity of the data. The challenge lies in developing a Shell script that provides a user-friendly interface, utilizes robust encryption algorithms, securely handles encryption keys or passwords, and can be easily integrated into existing workflows or used across different operating systems. The solution should address these concerns and provide a reliable and efficient means of file encryption, enhancing data security and privacy for both individuals and organizations.

1.3.2 Complex Engineering Problem

The following table must be completed according to your above discussion in detail. The column on the right side should be filled only on the attributes you have chosen to be touched by your own project.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Depth of knowledge required	Basic Programming Concepts,Bash Scripting,File Encryption Concepts,GNU Privacy Guard (GPG),File System Operations,Security Considerations,Operating System Basics.
P3: Depth of analysis required	File Encryption/Decryption Process, User Interaction and Input Validation, Security Considerations, Usability and Error Handling
P7: Interdependence	Bash scripting to implement the file encryption and decryption functionality. Understanding Bash scripting concepts and commands is essential for working with files, performing file operations (such as renaming, reading, and writing), and executing commands within the script.

1.4 Design Goals/Objectives

- Develop a user-friendly Shell script.
- Implement robust file encryption algorithms.
- Enable encryption key generation.
- Securely handle encryption keys or passwords.
- Support decryption.
- Ensure cross-platform compatibility.
- Enhance data security and privacy.

1.5 Application

- Individuals can use the Shell script to encrypt personal files, such as financial records, personal documents, or private photos, ensuring that their sensitive information remains secure in case of unauthorized access.
- Organizations can utilize the Shell script to encrypt confidential business data, such as financial statements, client information, or trade secrets.
- Many industries and jurisdictions have strict regulations regarding the protection of personal or sensitive data. By using the Shell script for file encryption, organizations can meet the requirements outlined by data protection laws, such as the General Data Protection Regulation.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

Files are an integral part of our daily lives. They contain a wide range of information, including personal documents, financial records, sensitive corporate data, and much more. Protecting the confidentiality and privacy of these files is crucial to prevent unauthorized access and maintain data integrity. File encryption and decryption are powerful techniques used to secure files by transforming their contents into an unreadable form and then restoring them to their original state.

2.2 Project Details

I am going to do a project which is encryption and decryption of a file in shell script. Here, we will create a text file then encrypt it using some lines of shell script code and lines of command of linux operating System. This project will take input which is user want to do either encryption or decryption. after taking choice it require a password. By giving this we will generate encrypted file.

2.3 Implementation

```
#!/bin/bash

echo "This is a simple file encrypter/decrypter"
echo "Please choose what you want to do"

choice="Encrypt Decrypt"

select option in $choice; do
    if [ $REPLY = 1 ]; then
```

```
        echo "Please enter the filename you want to encrypt"
        read file
        gpg -c -o temp_file.gpg $file
        mv temp_file.gpg $file
        echo "The file has been encrypted"
    else
        echo "Please enter the filename you want to decrypt"
        read file2
        gpg -d $file2 > temp_file
        mv temp_file $file2
        echo "The file has been decrypted"
    fi
    break
done
```

2.4 Algorithms

-

Algorithm 1: Sample Algorithm

```
if encryption then
    read file;
    gpg -c $file$
else
    read file2;
    gpg -d $file2$
else
    Do the rest
```

Chapter 3

Performance Evaluation

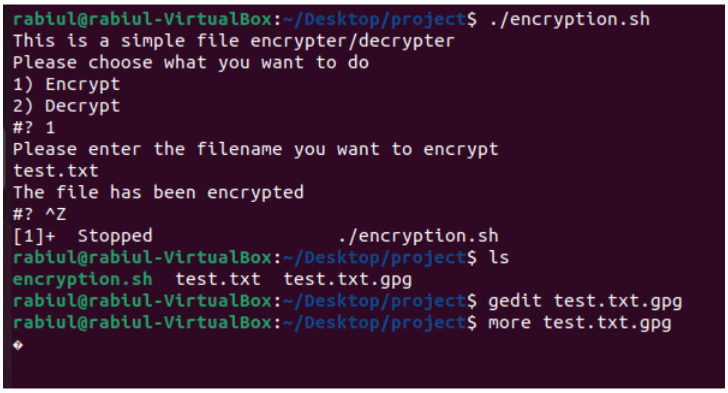
3.1 Simulation Environment/ Simulation Procedure

Here we are not using any specific IDE for implementation. We are using general nodepad to write the code and then run this using terminal with some linux command line.

3.2 Results Analysis/Testing

We have run this project in terminal. So output will shown there as given below:

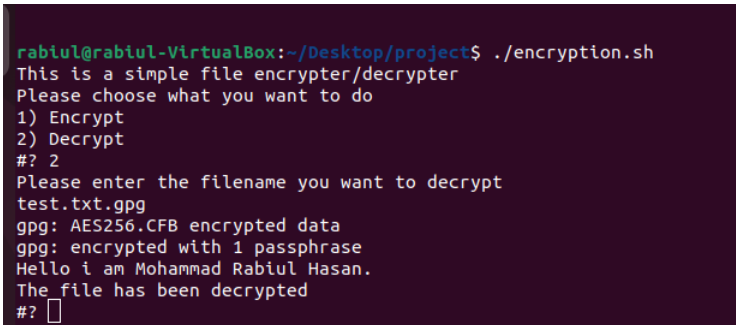
3.2.1 Encryption



```
rabiul@rabiul-VirtualBox:~/Desktop/project$ ./encryption.sh
This is a simple file encrypter/decrypter
Please choose what you want to do
1) Encrypt
2) Decrypt
#? 1
Please enter the filename you want to encrypt
test.txt
The file has been encrypted
#? ^Z
[1]+  Stopped                 ./encryption.sh
rabiul@rabiul-VirtualBox:~/Desktop/project$ ls
encryption.sh  test.txt  test.txt.gpg
rabiul@rabiul-VirtualBox:~/Desktop/project$ gedit test.txt.gpg
rabiul@rabiul-VirtualBox:~/Desktop/project$ more test.txt.gpg
�
```

Figure 3.1: Encryption

3.2.2 Decryption



```
rabiul@rabiul-VirtualBox:~/Desktop/project$ ./encryption.sh
This is a simple file encrypter/decrypter
Please choose what you want to do
1) Encrypt
2) Decrypt
#? 2
Please enter the filename you want to decrypt
test.txt.gpg
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
Hello i am Mohammad Rabiul Hasan.
The file has been decrypted
#? 
```

Figure 3.2: Decryption

3.3 Results Overall Discussion

The provided shell script is a simple file encrypter/decrypter that utilizes the GPG (GNU Privacy Guard) tool. It offers a straightforward command-line interface for users to encrypt or decrypt files. When executed, the script prompts the user to choose between encryption and decryption options.

If the user selects encryption, they are prompted to enter the filename of the file they want to encrypt. The script then employs the gpg -c command, using symmetric key encryption, to encrypt the file. The encryption process is carried out using the passphrase provided by the user during encryption. Once the encryption is completed, a message is displayed confirming the successful encryption of the file.

Alternatively, if the user selects decryption, they are prompted to enter the filename of the file they want to decrypt. The script utilizes the gpg -d command to decrypt the specified file. The decryption process requires the correct passphrase that was used during encryption. Following the decryption, a confirmation message is shown, indicating the successful decryption of the file.

To present a menu-like interface, the script employs a select statement, allowing the user to choose between encryption and decryption options. The selected option is stored in the option variable, which then determines the subsequent actions taken by the script.

3.3.1 Complex Engineering Problem Discussion

Here I have gather depth of knowledge about file encryption decryption method. Then I have analyse the required data to implement and how to implement it in eap[;?"tsier way and efficient.

Chapter 4

Conclusion

4.1 Discussion

File encryption is the process of converting the contents of a file into an unreadable form, known as ciphertext, using an encryption algorithm. The purpose of file encryption is to protect the confidentiality and integrity of the data stored in the file. Encryption ensures that even if the file is accessed by unauthorized individuals, they won't be able to understand its contents without the correct decryption key or password.

File decryption is the reverse process of encryption. It involves taking the encrypted ciphertext and applying the decryption algorithm, along with the correct decryption key or password, to convert the ciphertext back into its original plaintext form. Decryption is performed by authorized individuals who possess the correct decryption key, enabling them to retrieve and access the original file contents.

4.2 Limitations

Here I have implemented it for only txt types documents. It is not applicable for other types of decuments.

4.3 Scope of Future Work

By implementing additional security features to strengthen the encryption and decryption process. This could involve incorporating stronger encryption algorithms, adding password complexity requirements, or implementing two-factor authentication for key access.