

**COMPUTER SCIENCE**

**INTERNAL ASSESSMENT**

**Cypher Encryption**

**Shawn Grant**

**Campion College**

**6B**

**TABLE of CONTENTS**

* **Problem Definition**
  + Background 1
  + Context Description 2
  + Problem Description 3
  + Objective Of Solution 4
* **Structured Chart 5**
* **Narrative 6**
* **Record Definitions 7**
* **Pseudocode 8**
* **C code 9**
* **Code Test Plan 10**

**PROBLEM DEFINITION**

**BACKGROUND**

Privacy of information has been an important part of history, from messages been sent secretly by birds to now, where we use the internet to send most of our information. Information sent over the internet can include: usernames, passwords, addresses and telephone numbers**.** With the rise of the internet, new threats to data security have arrived. All computer systems are susceptible to intrusion by hackers. New technology has been developed to combat the threat of hackers, but no system is hacker-proof yet.

Everyday people send sensitive data over the internet via social media or email. Although these networks may be strong against computer based attacks, no thought is put into the fact that once your phone or laptop is left open anyone can simply open your messages and view the data you’ve sent to others.

**CONTEXT DESCRIPTION**

Encryption is the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot.

When people send information over the internet, messages are encrypted and sent over a network to the recipient’s router, where it is decrypted and returned to the recipient’s device and represented as a plain text or file.

This works well for ensuring that external hackers cannot get intelligible data from information sent over the network.

This encryption, however, does not protect against anyone with access to the device from viewing messages as the data is only encoded while it is in transit.

If, say, you text a friend or relative your email password (for whatever reason), and either of you gets robbed and your phone stolen. There’s no way of protecting that message that was sent. The thief now has your email. Even if there is a lock on the device, hackers have always figured out ways of cracking those, since they are normally not very complex.

If the message sent was in an encoded form it would add an extra layer of protection on the information. There would be no way of determining what the original message was.

**PROBLEM DESCRIPTION**

* We share sensitive information regularly over plain text. This could be anything from a password to credit card information.
* Our information is only secure while it is being sent over a trusted network, by email or social media application.
* The security of that information is comprised if someone simply takes your unlocked phone or computer (or you give it to them) and opens your email in a browser you’ve already signed in to.
* A stolen device is the easiest way to get access to someone’s information.

**OBJECTIVE OF SOLUTION**

* Allow users to encode any text message to a code using a cypher to change the text into an encrypted form.
* Ensure that sensitive information stored on a device locally is protected against theft of the device
* Allow users to send an encrypted message to anyone who has the application to decode it.
* Allow users to save the encoded text to a file for later use.

**STRUCTURED CHART**

**DRIVER**

**SplashScreen**

**Decode**

**SaveToFile**

**GoodBye**

**ShowHistorye**

**Clear**

**RemoveItem**

**strInsert**

**SaveToFile**

**Encode**

**strInsert**

**MainMenu**

**Login**

**strRemove**

**strIndexOf**

**strRemove**

**strIndexOf**

**NARRATIVE**

**DRIVER**

1. Display a splash screen with the program name
2. Call Login()
3. Display main menu: A) Encode B) Decode C) Open file D) Show History E) Exit
4. Prompt and accept choice
5. Call the function that corresponds to the choice using a switch

**LOGIN ()**

1. Prompt the user for username
2. Read the username

**ENCODE ()**

1. Prompt the user for any text to encode
2. Read the text
3. Encode the text by swapping characters with a predetermined corresponding symbol
4. Display the encoded text
5. Prompt the user to save the text to file or return to menu

**DECODE ()**

1. Prompt the user for previously encoded text to decode
2. Accept the text
3. Decode the text by swapping the encoded characters with the corresponding English character
4. Display the encoded text

**OPENFROMFILE ()**

1. Prompt the user for a file name of a local .cyph file
2. Read the filename
3. Attempt to locate the file in the program’s custom folder
4. Open the file and extract the text in it
5. Call Decode() on the text in the .cyph file : Decode (filename)
6. Display the text

**SHOWHISTORY ()**

1. Locate the local history.cyph file
2. Open the file and display the text inside
3. Prompt the user to delete an item from the history or clear all items

**GOODBYE ()**

1. Display goodbye message
2. Exit the program

**RECORD DEFINITION**

1. **TextCypher** Record 🡪 Contains two strings, storing the original text the user entered and the encoded version of the text

**Record TextCypher**

**original: String**

**encoded: String**

**reciever: String**

**EndTextCypher**

1. **User** Record 🡪 Stores the current logged in user

**Record User**

**username: String**

**password: String**

**EndmUser**

**PSEUDOCODE**

'data type to hold the before and after text of an encryption

Record **TextCypher**

original: String

encoded: String

reciever: String

EndTextCypher

'data type to hold the current user

Record **User**

username: String

password: String

EndUser

**Driver**()

userInput: TextCypher

user: User

choice: Character

LoginSignUp (user)

choice = MainMenu ()

While (choice <> 'E' AND choice <> 'e') DO

CASE of choice

Case 'A':

Case 'a':

Encode (userInput)

Case 'B':

Case 'b':

Decode (userInput)

Case 'C':

Case 'c':

OpenFromFile ()

Case 'D':

Case 'd':

ShowHistory ()

Default

Print "INVALID OPTION!"

EndCase

choice = MainMenu ()

EndWhile

EndDriver

**LoginSignUp** ( user: User)

nameOnFile, passwordOnFile: String

fp: FILE

If ((fp = Open File “UserData.dat” for reading and writing)) <> NULL) Then

Read nameOnFile from File, fp

Read passwordOnFile from File, fp

'check if to sign up or login

If (nameOnFile <> "") Then

'login

Repeat

Print "LOGIN"

Print "ENTER USERNAME: "

Read user.username

Print "ENTER PASSWORD: "

Read user.password

Until (user.username == nameOnFile AND user.password == passwordOnFile)

Else

'sign up

Repeat

Print "SIGN UP"

Print "ENTER USERNAME: "

Read user.username

Print "ENTER PASSWORD: "

Read user.password

Until (user.username <> "" AND user.password <> "")

'save to file

Write user.username, user.password to File fp

EndIf

Close File, fp

Else

Print "CANNOT LOAD USER DATA... RESTART THE PROGRAM"

EndIf

EndLoginSignUp

**MainMenu**(): Character

c: Character

Print "A) ENCODE A MESSAGE"

Print "B) DECODE A MESSAGE"

Print "C) NEW FROM FILE"

Print "D) SHOW HISTORY

Print "E) EXIT"

Print "CHOOSE AN OPTION ABOVE: "

Read c

return c

EndMainMenu

**C CODE**

/\*

Programmer : Shawn Grant

Date : 28,1, 2019

Purpose : Main driver module

Filename : main.c

\*/

#include <stdio.h>

#include <stdlib.h>

#include <windows.h>

#include <string.h>

#include <conio.h>

#include "CYPHER.H"

void LoginSignUp (User);

char MainMenu ();

int main()

{

TextCypher userInput;

User user;

char choice;

LoginSignUp(user);

choice = MainMenu();

while(choice != 'E' && choice != 'e')

{

switch(choice)

{

case 'A':

case 'a':

Encode(userInput);

break;

case 'B':

case 'b':

Decode(userInput);

break;

case 'C':

case 'c':

OpenFromFile();

break;

case 'D':

case 'd':

ShowHistory();

break;

default:

printf("INVALID OPTION!");

printf("Press any key to continue...");

getch();

break;

}

choice = MainMenu();

}

GoodBye();

return 0;

}

void LoginSignUp(User user)

{

char nameOnFile[25], passwordOnFile[25];

FILE \*fp;

if((fp = fopen("userData.dat", "w+")) != NULL)

{

fscanf(fp, "%s", nameOnFile);

fscanf(fp, "%s", passwordOnFile);

//check if to sign up or login

if(strcmp(nameOnFile, "") != 0)

{

//login

do

{

printf("LOGIN ");

printf("ENTER USERNAME: ");

CreateTextBox("%s", user.username, 0);

printf("ENTER PASSWORD: ");

CreateTextBox("%s", user.password, 1);

system("cls");

}

while(strcmp(user.username, nameOnFile) != 0 && strcmp(user.password, passwordOnFile) != 0);

}

else

{

//sign up

do

{

printf("SIGN UP ");

printf("ENTER USERNAME: ");

CreateTextBox("%s", user.username, 0);

printf("ENTER PASSWORD: ");

CreateTextBox("%s", user.password, 1);

system("cls");

}

while(strcmp(user.username, "") == 0 && strcmp(user.password, "") == 0);

//save to file

//fprintf(fp, “%s %s”, user.username, user.password);

}

fclose(fp);

}else

{

printf("CANNOT LOAD USER DATA... RESTART THE PROGRAM");

}

}

char MainMenu()

{

char c;

system("cls");

printf ("A) ENCODE A MESSAGE");

printf ("B) DECODE A MESSAGE");

printf ("C) NEW FROM FILE");

printf ("D) SHOW HISTORY");

printf ("E) EXIT");

printf("CHOOSE AN OPTION ABOVE: ");

c = getch();

return c;

}