

**COMPUTER SCIENCE**

**INTERNAL ASSESSMENT**

**Cypher Encryption**

**Shawn Grant**

**Cand. #: 1000162122**

**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 in plain text.

This is where local encryption is useful. Text and files can be encoded, so that even with access to the device it is impossible for anyone to make sense of the information. There are many different encryption algorithms, most popular being: MD5, RSA, SHA-1 and 3DES.

**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.
* Many persons don’t have a simple screen lock on their phone, removing the first layer of security on a device
* The security of that information is comprised if someone simply takes your unlocked phone or computer (or you give it to them) and opens the email or messaging application 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 an unreadable form
* Ensure that sensitive information stored on a device locally is protected against prying eyes
* Allow users to send an encrypted message and files to anyone without ever disclosing information.

**STRUCTURED CHART**

**DRIVER**

**SplashScreen**

**Decode**

**SaveToFile**

**GoodBye**

**ShowHistorye**

**Clear**

**RemoveItem**

**SaveToFile**

**Encode**

**MainMenu**

**Login**

**strAppend**

**strAppend**

**strPresentAtIndex**

**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) Update Login Info F) Exit

1. Prompt and accept choice
2. Call the function that corresponds to the choice using a switch

**LOGIN ()**

1. Open the userData.dat file
2. Read the text from the file
3. If file is empty  
   Prompt the user for username and password

Read the username and password

Save username & password to userData.dat file  
 user is logged in, continue

1. If file is **not** empty  
   Prompt the user for username and password

Read the username and password

If username & password is **equal to** username & password from userData.dat file  
 user is logged in, continue

**ENCODE ()**

1. Prompt the user for any text to encode
2. Read the text
3. Loop through the text and change each letter to its ascii code
4. Append the ascii code to the new encoded text
5. Display the encoded text
6. Update the history file with the new cypher
7. 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. Loop through the text and change each 3 digit ascii code to its character value
4. Append the character to the new decoded text
5. Update the history file with the cypher
6. 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
4. Open the file and read 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

**ADDTOHISTORY( TextCypher)**

1. Locate/create the local history.cyph file
2. Append the textcypher to the file

**GOODBYE ()**

1. Display goodbye message
2. Exit the program

**RECORD DEFINITION**

1. **TextCypher** Record 🡪 Contains information about the encoding

**Record TextCypher**

**original: String**

**encoded: String**

**reciever: String**

**dateTime: String**

**EndTextCypher**

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

**Record User**

**username: String**

**password: String**

**EndmUser**

**PSEUDOCODE**

---See Pseudocode.pdf---

**C CODE**

---See zip file---

**CODE TEST PLAN**

Input values and their expected output results, to test that the program works.

**Login**

|  |  |
| --- | --- |
| Username | Password |
| shawn | pwd123 |

**Encode**

|  |  |  |
| --- | --- | --- |
| Menu choice | English Text (input) | Encoded Text (output) |
| ‘A’ or ‘a’ | “Hello World” | 2080109116116119|095119122116108 |

**Encode**

|  |  |  |
| --- | --- | --- |
| Menu choice | Encoded Text (input) | English Text (output) |
| ‘B’ or ‘b’ | 2080109116116119|095119122116108 | “HelloWorld” |