ASSIGNMENT 5 DESIGN DOCUMENT

Pehara Vidangamachchi

October 28th 2022

Ethan L. Miller

About:

This assignment aims to create a program that makes a key generator called keygen, an encryptor called encrypt and finally a decryptor called decrypt. Each program will have its own purpose to serve, such as keygen which is responsible for generating a public and private RSA key. Encrypt on the other hand will encrypt files with a public key and Decyprt will decrypt the file using its private key. We will be tasked with using mathematical formulas and the given pseudo code to implement these functions into our own program that executes the directions given.

Design Process for decrypt.c

The purpose of this program is to create a function decrypt function that is able to through several parse-options. It is known as the process of decoding encoded data. Decrypt is a function used to revert an encrypted message back into something readable without sacrificing safety. This is done with the use of keys, private and public which assist with the process. For decryption, the private key is used to translate the message into something readable once again. I aim to achieve this by using a multitude of for and if statements to help create a functioning program.

#include any header files

Create num function:

Initialize any variables

Create a for statement:

Set variable equal to the math function

Return the math

Create decrypt function:

Initialize any variables

Using a for loop set the parameters:

Use an if statement to further get into depth for decryption and check what character is being used

Create an if statement to decrypt a message:

Use the math to decrypt a message

Return the decrypted message

Design Process for encrypt.c

The purpose of this program is to create an encrypt function that is able to encrypt a message and additionally parse through several different options. Encryption is the process of encoding a given set of data and converting it to what's known as cypher text. Encryption is a function that takes a message and encrypts it with a public key so it's only readable if a private key is used to decrypt it. With the assistance of keys, the program is able to create an encrypted function alongside a mathematical formula to help translate its method of encryption. In order to achieve this I will use a multitude of if statements for loops to create a working program.

#include any header files

#define any variables

Create a main program for encryption:

Initialize any variables

Create a for loop to calculate the frequency:

Use the mathematical formula

Create a for loop after calculating the frequency:

Followed by an if statement and its mathematical counterparts:

Increment by a variable

Else:

Math formula for if you decrement it by a variable

Return the count

Design Process for numtheory.c

This program serves the purpose of containing the correct implementation of the number theory function. This will contain functions such as modular exponentiation and modular inverses which will help with the math behind encryption, decryption and key generation. Following what's given in the assignment, I can use the given pseudo-code to translate that over to my own interpretation of the pseudo-code.

#include any header files

#define any variables that'll be used

Create a modular exponentiation function:

Initialize any variables

Create a for statement with the parameters:

Create a while statement with parameters:

Create an if statement for odd:

Create a math formula

Include two more math functions

Return the variable

Create the miller Rabin function:

Initializer nay variables

For loop with parameters:

Set the range and its math counterpart

If statement:

For when y is one or less than n

Set a variable equal to one

Create a while loop with its respective range:

Include the math following this

If the statement in case of variable is logically equal to one:

Return false

Math for more variables

If statement for if the variable isn't equal:

Return false

Return true

Create modular inverses function:

While loop to check if cognition is not equal to zero:

Set variables equal to another variable

Return variable

Create a function for the mod inverse:

Initialize all important variables

Set variables equal to their equations

Create a while loop for when the variable is not equal to zero:

Include basic math formulas that are given

If statement when greater than one:

Return no inverse

If statement is less than zero:

Return variable

Design Process for randstate.c

This file contains the implementation of the random state interface used for the RSA library and the number theory functions for my code. This file is essential to using RSA public and private keys. This file helps bring all the necessary code into one compilable program.

#include any header files

 # initialise any important variables

Create the main function for Randstate:

Create a while loop to run Randstad code:

Run with the given parameters

Return

Design Process for rsa.c

This program will contain the implementation of the RSA library that will be used within my program. The file RSA.c is the library for RSA that I can utilise within my code. This helps to specify certain command line options, such as the minimum bits and the number of miller Rabin iterations.

#include any header files

 # initialise any important variables

Create the main function for rsa.c:

Using the library setup each function

Create different functions for each library function

Execute and implement math functions

Return

Design Process for keygen.c

This program will contain the implementation of the RSA library that will be used within my program. Using this program helps create keys that will be utilized within other segments of the code to create a random key to help encrypt and decrypt the sequence.

#include any header files

 # initialise any important variables

Create the main function for keyegn.c:

Create a while loop to run the key generation:

Create an if statement to check the parameters:

Continue running the program with is math function

If statement if it is outside of range

Return False

Return