Assignment 5

Public Key Cryptography

DESIGN.pdf

Reuben T. Chavez

February 17, 2022

Pseudocode

```
#Libaries
import randstate
import numtheory
import rsa
import stdlib
import bool
import stdint

# Psuedocode for keygen.c

## Usage ##
function usage is
    input: executable
    output: void

    print(
    "Synopsis of Keygen\n"

    "Usage of KeyGen\n"

    "Options for KeyGen\n"
    )

function main is
    input : argument count argc and argument vector argv
    output: zero to exit program

    opt <- 0
    Set bits as a unsigned 64 bit number
    verbose <- false
    iterations <- 50
    Set public rsa file to be read
    Set private rsa file to be read
    Set seed to explicit starting point
    Set random to the created seed
    while getting commands from command line do
        switch command:
            case bits:
                set bits to the user's argument
                break
            case iteration :
```

```
            set iteration to the command line arguments
            break
        case public file :
            if given files exists
                pbfile <- users's argument
                break
            Print that the given file does not exist and end
                program
        case private file :
            if given files exists
                pvfile <- users's argument
                break
            Print that the given file does not exist and end
                program
        case seed
            seed <- User's argument
            Inintialize reandom to staart at given seed
            break
        case verbose:
            verbose <- true
            break
        default help:
            prints usage and ends program

Check the both the public file and private file have file key
    permisiion to 600

Ininitialize random state with given seed

Create public key with function in rsa library

Create private key with function in rsa library

Get current user's name in the /home/username path

Convert username to integer of base 62 and use rsa sign in
    library

if verbose is true:
    print {Username
           Signature
           First Large Prime
           Second Large Prime
           Public Modulus
           Public Exponent
```

```
              Private Key
               }
    Write public modulus , public exponent, siganuture , and
        username into public file

    Write public modulus, private key into private file

    Clear all given files and mpz intergers
    return 0
```

```
#Libaries
import numtheory
import rsa
omport randstate
import stdlib
import bool
import stdint

# Psuedocode for encrypt.c

## Usage ##
function usage is
    input: executable
    output: void

    print(
    "Synopsis of encrypt\n"

    "Usage of encrypt\n"

    "Options for encrypt\n"
    )

function main is
    input : argument count argc and argument vector argv
    output: zero to exit program

    intialize opt to 0
    initialize input to standard input
    initialize output object to standard output
    initialize pvfile object to private file
    initalize verbose as false
```

```
while getting commands from command line do
    switch command:
        case i:
            if file exists:
                input is set to read file
                break
            print that the file does not exist
            stop running
        case o :
            if file exists:
                output is set to read file
                break
            print that the file doe not exist
            stop running
        case n :
            if file exists:
                pvfile is set to read file
                break
            print that the file doe not exist
            stop running
        case verbose:
            set verbose to true
        default help:
            prints usage and ends program

Initalize mpz-t varibles that store the public modulus and
    public exponent

Read given file the set the private key and public modulus

if verbose:
    print(
         The public modulus
         The public exponent
         )
decrypt the give input file to the given output file with the
    public modulus and public exponent

clear the mpz-t varibles
close all opened files
return 0
```

```
#Libaries
```

```
import numtheory
import rsa
import randstate
import stdlib
import bool
import stdint

# Psuedocode for decrypt.c

## Usage ##
function usage is
    input: executable
    output: void

    print(
    "Synopsis of decrypt\n"

    "Usage of decrypt\n"

    "Options for decrypt\n"
    )

function main is
    input : argument count argc and argument vector argv
    output: zero to exit program

    intialize opt to 0
    initialize input to standard input
    initialize output object to standard output
    initialize pvfile object to private file
    initalize verbose as false

    while getting commands from command line do
        switch command:
            case i:
                if file exists:
                    input is set to read file
                    break
                print that the file does not exist
                stop running
            case o :
                if file exists:
                    output is set to read file
                    break
                print that the file doe not exist
                stop running
            case n :
                if file exists:
                    pvfile is set to read file
                    break
                print that the file doe not exist
                stop running
            case verbose:
                set verbose to true
            default help:
                prints usage and ends program

    Initalize mpz-t varibles that store the public modulus and
        public exponent
    Initilez username to NULL

    Read given file the set the public modulus, public exponent,
        signature, and private key

    if verbose:
        print(
             The username
             The signature
             The public modulus
             The public exponent
             The private key
             )
    encrypt the give input file to the given output file with the
        public modulus and public exponent

    clear the mpz-t varibles
    close all opened files
    return 0
```

```
#Pseudocode for rsa.c


#Libaries
import numtheory
import rsa
import randstate

import stdbool
import stdint
import stdio
import stdlib
import gmp
import gmp
import math


function lcm is :
    input: mpzt output, mpzt a mpzt b
    output nothing

    Initialize varibles for numerator and denominator

    Set numertor to equal to absult value of the a times b

    Set denominator to the greatest common divisior of a and b

    Set output to numerator divided by denominator

    clear mpzt varibles

function rsa-make-pub
    input: mpzt p, mpzt, q , mpzt n. mpzt e, nbits , iterations
    output: nothing

    make p equal to prime number that is in range of nbit/4 to
        3*nbits/r
    make q equal to rest of the nbits
```

```
#Pseudocode for randstate.c
```

```
#Pseudocode for numbtheory.c
```