asgn5

Ryan Hui

rhui1

Design

Make a key generator, encryptor, and decryptor for the Schmidt-Samoa algorithm. The mathematics part of the algorithm will be implemented with the numtheory.c file which incldues gcd, mod inverse, pow mod, is prime, and make prime functions. The output of asgn5 should be a decrypted file with text.

Files

-decrypt.c
-encrypt.c
-keygen.c
-numtheory.c
-numtheory.h
-randstate.c
-randstate.h
-ss.c
-ss.h
-Makefile
-README.md
-DESIGN.pdf
-WRITEUP.pdf

Pseudocode

numtheory.c

gcd function
while b doesn't equal 0
t equals b
b equals a mod b
a equals t
return a

mod inverse function
r equals n
r' equals a
while r' doesn't equal 0

q equals r/r'
r equals r'
r' equals r - q * r'
t equals t'
t' equals t- q * t'

if r greater than 1
return no inverse

if t less than 0
t equals t + n
return t

pow mod function
v equals 1
p equals a
while d greater than 0
if d is odd
v equals v * p mod n
p equals p * p mod n
d equals d / 2
return v

is prime function
n - 1 = 2 to the power of s * r such that r is odd

for 1 to k
choose random number a in range of (2,3,..,n-2)
y equals power mod(a,r,n)
if y doesn't equal 1 and y doesn't equal n - 1
j equals 1
while j less than or equal to s - 1 and y doesn't equal n - 1
y equals pow mod(y,2,n)
if y equals 1
return false

j equals j + 1

if y doesn't equal n - 1
return false

return true

make prime function
while value is false
generate random number for temp
if is prime(temp, iters)
value equals true

ss.c

ss make pub function
create prime p and q using make prime function
decide number of bits in random number in the range
n equals p * p * q
obtain random number and check p != q -1 and q != p - 1

ss write pub function
write public ss key to pbfile

ss read pub function
reads ss key from pbfile
read the value which should be a hexstring

ss make priv function
create a new ss private key with primes p and q and public key n
ans equals gcd(p-1, q-1)
temp equals p - 1 * q - 1
pq equals p * q
lambda equals temp / ans
d equals mod inverse(n, lambda)

ss write priv function
writes ss key to pvfile
format should be a hexstring

ss read priv function
read private ss key from pvfile
format should be a hexstring

ss encrypt function
performs ss encryption by computing ciphertext c
encrypt message m using public key n
E(m) = c = mod (n)

ss encrypt file function
encrypts contents of infile
write encrypted content to outfile
data in infile should be encrypted in blocks
value of block cannot be 0
value of block cannot be 1

calculate block size k
dynamically allocate array that can hold k bytes
set zeroth byte of block to 0xFF

while still unprocessed bytes in infile
read at most k - 1 from infile
covert the read bytes to mpz
encrypt m with ss encrypt and write number to outfile as hexstring

ss decrypt function
performs ss decryption
compute message m by deciphering ciphertext c
use private key d and public modolus n

ss decrypt file function
decrypts content of infile
write decrypted contents to outfile
dynamically allocate an array that can hold k bytes

iterate over infile
scan in hexstring
convert c back into bytes
write out j - 1 bytes from index 1 of the block to outfile

randstate.c

randstate innit function
set random seed
initialize state
set state to seed

randstate clear function
clear state

keygen.c

main function
get opt
-b: specifies min bits
-i: specifies iterations
-n: specifies public key file
-d: specifies private key file
-s: specifies random seed
-v: enables verbose output
-h: displays program synopsis

parse command line
open public and private key files
set private key file permissions to 0600
initialize random state
make public and private keys using ss make pub
get currrent user's name as a string
write computed public and private key to files
if verbose output is enabled
print username, first large prime p, second large prime q, public key n, private exponent d, privvate modulus pq

close private and public key files

encryptor.c

main function
get opt
-i: specifies input file
-o: specifies output file
-n: specifies public key
-v: enables verbose output
-h: displays program usage

parse command line
open public key file
read public key
if verbose output is enabled
print username, public key n
print all bit information

encrypt file using ss encrypt file function
close public key file and clear mpz

decryptor.c

main function
get opt
-i: specifies input file
-o: specifies output file
-n: specifies file containing private key
-v: enables verbose output
-h: displays program usage

parse command line
open private key
read private key
if verbose output enabled
print private modulus pq, private key d

decrypt file using ss decrypt file function
close private key file and clear mpz

Structure

-numtheory.c functions include gcd, mod inverse, pow mod, is prime, and make prime. These functions utilize mpz variables to calculate the desired mathematical or boolean outcomes and are used throughout assignment 5 ss.c functions.

-ss.c functions include make pub, make priv, write pub, write priv, read pub, read priv, encrypt, encrypt file, decrytp, and decrypt file. These functions implement the actual S.S encryption algorithms as well as what prints the encrypted and decrypted messages to the files.
-randstate.c functions include randstate init and randstate clear and are used to create and clear the random states used in numtheory.c

-keygen.c is a file that implements the generation of the private key. This program has multiple options that can be specified.

-encrypt.c is a file that implements the encryption of the message. This program has multiple options that can be specified.

-decrypt.c is a file that implements the decryption of the message. This program has multiple options that can be specified.

Credit

-cse13s discord
-piazza