|  |  |
| --- | --- |
|  | from pip.\_vendor.distlib.compat import raw\_input |
|  | from pip.\_vendor.idna import unichr |
|  |  |
|  | try: |
|  | input = raw\_input |
|  | except NameError: |
|  | pass |
|  | try: |
|  | chr = unichr |
|  | except NameError: |
|  | pass |
|  | p1=int(input('Enter prime number1: ')) |
|  | p2=int(input('Enter prime number2: ')) |
|  | print("Choosen primes:\np1=" + str(p1) + ", p2=" + str(p2)) |
|  | print("\n") |
|  | n=p1\*p2 |
|  | print("n = p1 \* p2 = " + str(n) ) |
|  | print("\n") |
|  | phi=(p1-1)\*(p2-1) |
|  | print("Euler's function (totient) [phi(n)]: " + str(phi)) |
|  | print("\n") |
|  |  |
|  | def gcd(a, b): |
|  | while b != 0: |
|  | c = a % b |
|  | a = b |
|  | b = c |
|  | return a |
|  | def modinv(a, m): |
|  | for x in range(1, m): |
|  | if (a \* x) % m == 1: |
|  | return x |
|  | return None |
|  | def coprimes(a): |
|  | l = [] |
|  | for x in range(2, a): |
|  | if gcd(a, x) == 1 and modinv(x,phi) != None: |
|  | l.append(x) |
|  | for x in l: |
|  | if x == modinv(x,phi): |
|  | l.remove(x) |
|  | return l |
|  | print("Choose from array:\n") |
|  | print(str(coprimes(phi)) ) |
|  | print("\n") |
|  | e=int(input()) |
|  | d=modinv(e,phi) |
|  | print("public key is (e=" + str(e) + ", n=" + str(n)) |
|  | print("\n") |
|  | print("private key is (d=" + str(d) + ", n=" + str(n)) |
|  | print("\n") |
|  | def encrypt\_block(m): |
|  | c = modinv(me, n) |
|  | if c == None: print('No multiplicative inverse ' + str(m) + '.') |
|  | return c |
|  | def decrypt\_block(c): |
|  | m = modinv(cd, n) |
|  | if m == None: print('No multiplicative inverse ' + str(c) + '.') |
|  | return m |
|  | def encrypt\_string(s): |
|  | return ''.join([chr(encrypt\_block(ord(x))) for x in list(s)]) |
|  | def decrypt\_string(s): |
|  | return ''.join([chr(decrypt\_block(ord(x))) for x in list(s)]) |
|  | s = input("Enter a message to encrypt: ") |
|  | print("Plain message: " + s ) |
|  | print("\n") |
|  | enc = encrypt\_string(s) |
|  | print("Encrypted message: "+ enc ) |
|  | print("\n") |
|  | dec = decrypt\_string(enc) |
|  | print("Decrypted message: " + dec ) |

**DM-Project-RSA-CRYPTOGRAPHY-Encryption-and-Decryption**

**Group Member's Names:**

1. Ahmed Mirza (63254) **LEADER**
2. Zain Ali Shakeel (62777)
3. **Project Description:**

Our project is build on Python 3.0. Our project is increption and decrption by using RSA Cryptography with crypher key (private and public both). Modern cryptosystems are typically classified as either public-key or private-key. Private-key encryption methods, such as the Data Encryption Standard(DES), use the same key to both encrypt and decrypt data. The key must be known only to the parties who are authorized to encrypt and decrypt a particular message. Public-key cryptosystems, on the other hand, use different keys to encrypt and decrypt data. The public-key is globally available. The private-key is kept confidential.

# DM 103348: RSA CRYPTOGRAPHY-ENCRYPTION-AND-DECRYPTION

## PROJECT MEMBERS

| **StdID** | **Name** |
| --- | --- |
| **63254** | **Ahmed Mirza (Leader)** |
| **62777** | **Zain Ali Shakeel** |

## Project Description

The aim of our project it is used for security purpose.In RSA CRYPTOGRAPHY both public and private key are used for encrypt a message. our project logic is that the two large prime number are mulitiply and Multiplying these two numbers is easy to determining the original prime number two large number n and n1 then take their modulus.

## Discrete Math Concepts Used:

* Gcd
* Prime number
* Phi function
* Key
* Modulus

### GCD:

In discrete mathematics, the greatest common divisor (gcd) of two or more integers, which are not all zero, is the largest positive integer that divides each of the integers. For example, the gcd of 8 and 12 is 4.

### Prime number:

Prime numbers are the building blocks of the integers. A prime number is a positive integer greater than one that has only two divisors: 1, and the number itself. For example, 17 is prime because the only positive integers that divide evenly into it are 1 and 17.

### Phi function:

Definition 3.8. 1 ϕ(n) is the number of non-negative integers less than n that are relatively prime to n. In other words, if n>1 then ϕ(n) is the number of elements in Un, and ϕ(1)=1.

### Key:

A key is used to identify the number of categories present in a graph. It is also called a legend. A key on a pictograph tells us how many each picture stands for. Look at the following pictograph.

### Modulus:

In mathematics, the absolute value or modulus of a real number x, denoted |x|, is the non-negative value of x without regard to its sign. Namely, |x| = x if x is positive, and |x| = −x if x is negative, and |0| = 0. For example, the absolute value of 3 is 3, and the absolute value of −3 is also 3.

## Example 1: By using Python Language:

The language we are using for our project is Python. This is a language we just started learning and we are quite familiar with it.

s = input("Enter a message to encrypt: ")

print("Plain message: " + s )

print("\n")

enc = encrypt\_string(s)

print("Encrypted message: "+ enc )

## Problems Faced:

* Lack of sources due to lockdown and quarantine.
* Loadshedding and Wifi problems leading towards the next problem.
* Cordination with members.
* Difficulty of understanding code.
* Logic issue.

## Problem 1: Difficulty of understanding code:

There is a reason we tell people to start with a "Hello, world." Because at that point in their journey, accomplishing that means that have conquered a certain number of fundamentals. You understand how to compile, how to run, how to call a function and pass it parameters. Writing a clone of an existing program is another big step, almost as important as the hello world step, only at a different stage of your journey. When you are done with it, you have overcome certain challenges. You had to figure out where to start, how to plan it, how to organize it, how to deal with the bugs, give it a personal touch, and package it up at the end so it is a usable thing. Even when cloning a simple program, you are learning all of the process.

For us it was difficult because most of us don't exactly know what cryptography is let alone know how it works and working on it was beyond our imagination. But there is always a start to something but it would've been better if it was explained to us at the beginning so it would've been easy for us to work on it. With quarantine it was extremely hard sometime problems are solved by contacting other people and coming face to face with one another. But like this it was hard to even contact each other at times.

## Problem 2: Lack of sources due to lockdown and quarantine:

When the univerisity was open it was easier for everyone to reach out and ask for a help understanding something, it could've been teacher or any senior who has done this course but due to lockdown it was hard to find any source, or anyone to help us out understand it. Of course the internet full of it but that does not mean it satisfy you with solution you are looking for. Happens to the best of us. Which made us pretty much crazy over what cryptography actually is.

## Problem 3: Loadshedding and Wifi problems leading towards the next problem:

Since we all live in Pakistan it's not even raise hand on this problem because we all face it. We all have faced load shedding at the worse time and at times that it even forgets to come back and that's pretty much the same with the wifi. Of course we all have data at times(not all the time, not to forget it's qauarentine and recharging balance is pretty hard now with shops closed and stuff) but it's useless when it comes to programming since it's pretty useless because you can't program with your phone.

## Problem 4: Cordination with members:

As I said due to lack of contact in person it was hard for all of us to reach out each other at times and loadshedding and net failure never helped either. So, coming to terms to create anything was extremely hard for all of us. Not saying that happens to everyone and others might have done work with it too but this was something that all of us faced and somehow, we also completed our project too.

## Problem 5: Logic issue:

Everyone have different way of approach and solving problems and it's good but when there is lack of co-ordination it gets hard even for developers to make something and we the students who are just learning what cryptography even is so, Logic one of the issue we all faced.

## References:

* Link Reference: (<https://www.comparitech.com/blog/information-security/rsa-encryption/>).
* Book Refrences (Rosen Discrete Mathematics and Its Application &th edition).
* Class Lecture.