RSA Algorithm

Aman Kumar Nirala (github.com/amannirala13)

Sunday, 6 June 2021

```
from random import seed
import random
import sys
from random import randint
import time

def is_prime(x):
    count = 0
    for i in range(int(x/2)):
        if x % (i+1) == 0:
            count = count+1
    return count == 1

def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def multiplicative_inverse(e, phi):
    for x in range(1, phi):
        if (e * x) % phi == 1:
            return x
    return None

def generate_keypair():
    random.seed(time.time())
    p = random.randint(0,1000)
    q = random.randint(0,1000)
    while True:
        if is_prime(p) and is_prime(q) and p != q:
            break;
        p = random.randint(0,1000)
        q = random.randint(0,1000)
    n = p * q
    phi = (p-1) * (q-1)
    e = random.randrange(1, phi)
    g = gcd(e, phi)
    while g != 1:
        e = random.randrange(1, phi)
        g = gcd(e, phi)
    d = multiplicative_inverse(e, phi)
    return ((e, n), (d, n))

def encrypt(pk, plaintext):
    key, n = pk
    cipher = [(ord(char) ** key) % n for char in plaintext]
    return cipher
```

```
def decrypt(pk, ciphertext):
    key, n = pk
    plain = [chr((char ** key) % n) for char in ciphertext]
    return ''.join(plain)
```

```
# HOW TO USE
public, private = generate_keypair()
message = input("Enter String to encrypt")
encrypted_message = encrypt(private, message)
print (''.join(map(lambda x: str(x), encrypted_message)))
decrypted_message = decrypt(public, encrypted_message)
print(decrypted_message)
```

```
Enter String to encrypt I love cryptography


11491711510311727322295101671181531151031734935331125924379511816222295920889353333535437957464511259
I love cryptography
```