Lab Report 2 Date:2081/03/28

Experiment 1: RSA

Title: Programming to learn about RSA.

1. Objective:The objective of this Lab-Work is to provide a secure communication over an insecure network, such as the internet which it achieves through public-key.
2. Theory: RSA is a widely used public-key cryptography system that enables secure data encryption and digital signatures. RSA is based on the mathematical difficulty of factoring large prime numbers. It is widely used for securing sensitive data, such as in online banking, email encryption,and other secure communications.

**How RSA Works:**

1. Key Generation:

* Two large prime numbers p and q are selected. Their product n=pxq is computed where n is used as modulus foe both the public and private keys.
* A public exponent e is chosen, which is typically a small value. It must be co-prime to (p-1)x(q-1).
* The private exponent d is calculated such that : d ≡ e-1 mod Φ(n).

1. Encryption:

* The sender uses the recipients public key (n,e) to encrypt the message.
* First, the plain-text message M is a represented as an integer m were 0<=m<n.
* The cipher text c is computed as: m=cemod n.

1. Decryption:

* The recipient decrypt the message using their private key(n,d).
* To decrypt, they compute: m=cd mod n.

2.1. Application of RSA

* Secure Communication
* Digital Signature
* Key Exchange
* Secure Email
* Digital Certificate
* Cryptographic Tokens
* Secure Software Distribution

1. Implementation of RSA.

*import random*

*import math*

*def is\_prime(number):*

*if number<2:*

*return False*

*for i in range (2,number//2+1):*

*if number%i==0:*

*return False*

*return True*

*def generate\_prime(min\_value,max\_value):*

*prime=random.randint(min\_value,max\_value)*

*while not is\_prime(prime):*

*prime=random.randint(min\_value,max\_value)*

*return prime*

*def mod\_inverse(e,phi):*

*for d in range(3,phi):*

*if(d\*e)%phi==1:*

*return d*

*raise ValueError("mod inverse doesnot exsist")*

*p,q=generate\_prime(100,200),generate\_prime(100,200)*

*while p==q:*

*q=generate\_prime(100,200)*

*n=p\*q*

*phi\_n=(p-1)\*(q-1)*

*e=random.randint(3,phi\_n-1)*

*while math.gcd(e,phi\_n)!=1:*

*e=random.randint(3,phi\_n-1)*

*d=mod\_inverse(e,phi\_n)*

*print("Public key:",e)*

*print("Private key:",d)*

*print("n :",n)*

*print("Phi of n:",phi\_n)*

*print("p:",p)*

*print("q:",q)*

*message="Hello world"*

*message\_encoded=[ord(ch) for ch in message]*

*cipher\_text=[pow(ch,e,n) for ch in message\_encoded]*

*print("The cipher text of the message is: ")*

*print(cipher\_text)*

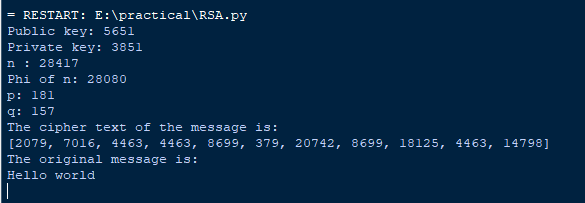
*message\_encoded=[pow(ch,d,n) for ch in cipher\_text]*

*message1="".join(chr(ch) for ch in message\_encoded)*

*print("The original message is: ")*

*print(message1)*

Output:



Conclusion: In the above page we have learned about RSA encryption and decryption and implemented them by using python.