udp Escuela de Informática y Telecomunicaciones

Tarea 4

---

criptografía y seguridad en redes

RSA

Nombre:José Arteaga
Profesor:Nicolás Boettcher
Ayudante:Nicolás Pino
Fecha de entrega:22-06-2020

Introduction:

Para esta tarea se escogió la segunda opción, la cual consiste en generar un script con el algoritmo RSA de cifrado asimétrico, con el cual se obtendrán las llaves públicas y privadas, para cifrar y descifrar mensajes respectivamente.

Luego de desarrollados los script se probarán con los link datos en el enunciado para corroborar si los script están correctamente desarrollados.

Finalmente se pide analizar los paquetes de postman enviados a dirección del enunciado para determinar si es posible obtener las llave públicas retornadas.

Explicacion del codigo:

```
def RSA():
    print("Ingrese dos números primos para generar sus llaves")

    while True:
        p=int(input("ingrese p: "))
        q=int(input("ingrese q: "))

        if ValidarPrimo(p) == False or ValidarPrimo(q) == False:
            print("Uno de los numeros ingresados no es primo , porfavor intente otra vez .....")
        else:
            break
```

El primer script es para obtener las llaves públicas y privadas de RSA a partir de dos números primos. Lo primero del código desarrollado es pedir dos números primos, los cuales serán las bases para generar las llaves públicas y privadas. Se pide ingresar p y q respectivamente y luego se valida si estos dos números son primos, para lo cual se utiliza una función declarada más arriba la cual es la siguiente :

```
def ValidarPrimo(numero):
    contador=0
    for i in range(1,numero+1):

        if (numero%i) == 0:
            contador +=1
    if contador == 2:
        return True
    else:
        return False
```

La cual compara todos los números de 1 hasta el número ingresado a la función y por cada múltiplo encontrado suma 1 al contador y si solo encontró dos múltiplos (1 y el propio número) retorna True, de lo contrario retorna False. Si la validación retorna algún False se vuelve a pedir ingresar número primos.

Siguiendo con el código se establece un n que es la multiplicación de los número primos p y q, y luego se setea un fi el cual es la multiplicación de (p-1)*(q-1), el cual nos ayudará a establecer la llave pública y privada, que son e y d respectivamente.

```
n = p*q
fi= (p-1)*(q-1)
e=0
d=0
```

Para establecer la llave e se deben cumplir dos condiciones la primera es que 1<e<fi y la segunda es que e debe ser coprimo de n y phi.

Para probar si e es coprimo con los números n y phi se utiliza la función de gcd() de la librería math de python, la cual entrega los máximos común divisor entre dos números, y para que sean coprimos el mcd debe ser igual a 1.

```
#escoger un e que sea coprimo con fi y con n , y debe ser menor que fi
for i in range(2, fi):
    if gcd(i, fi) == 1 and gcd(i, n) == 1 and i<fi:
        e = i
        break
```

y para que cumpla la primera condición limitamos el ciclo desde 2 hasta fi.

Finalmente para obtener la llave privada se debe encontrar un número d tal que cumpla la siguiente condición: (d*e)mod (fi) = 1, para esto se utiliza un ciclo for para encontrar dicho número que cumpla esta condición, el ciclo va desde 1 hasta un número muy grande para asegurarnos de encontrar dicho número. y la condición de que d debe ser distinto de e es para dar mayor seguridad al algoritmo, y por que sin esta condición el valor de d y e pueden llegar a ser iguales.

```
#escoger un d que cumpla la condicion de "(D*E)%mod(fi)=1"
for x in range (1,123456789):
    if (x*e)%(fi) == 1 and x!=e:
        d = x
        break
```

Luego de obtenidas las llaves el script imprime las están entregando las llaves privadas y públicas para esos números escogidos para cifrar.

```
print("Las llaves públicas son: E =",e,"N =",n," y las privadas son: D =",d," y N =",n)
```

Algoritmo para cifrar y descifrar

El script de cifrado de rsa nos pide 3 cosas las cuales son los dos valores de la llaves públicas y el valor del mensaje que queremos cifrar:

Luego se cifra con la siguiente fórmula: C=(m^e)%n

```
print("Introduzca los siguientes datos para encriptar su mensaje: ")
n = int(input("Valor de n: "))
e = int(input("Valor de e: "))
m = int(input("Valor de m: "))
crifrado= (m**e)%n
print("Su mensaje cifrado es: ",crifrado)
```

Y el script para descifrar nos pide 3 cosas que son una de las llaves públicas , la llave privada y el mensaje cifrado.

Luego se descifra con el siguiente algoritmo : m=(c^d)%n

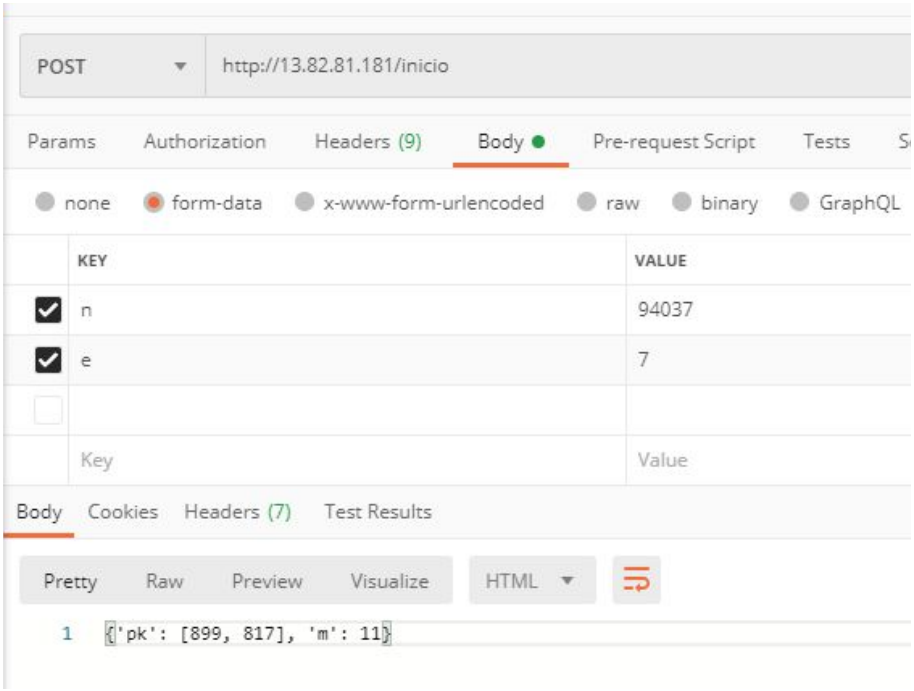
```
print("Introduzca los siguientes datos para desencriptar su mensaje: ")
n = int(input("Valor de n: "))
d = int(input("Valor de d: "))
c = int(input("Valor de mensaje cifrado: "))
descrifrado= (c**d)%n
print("Su mensaje descifrado es: ",descrifrado)
```

Comprobación por url:

para esto se escogen dos números primos, en este caso escogí p=347 y q=271.

```
Ingrese dos números primos para generar sus llaves
ingrese p: 347
ingrese q: 271
Las llaves públicas son: E = 7 ,N = 94037  y las privadas son: D = 53383 ,P = 347 ,Q = 271
```

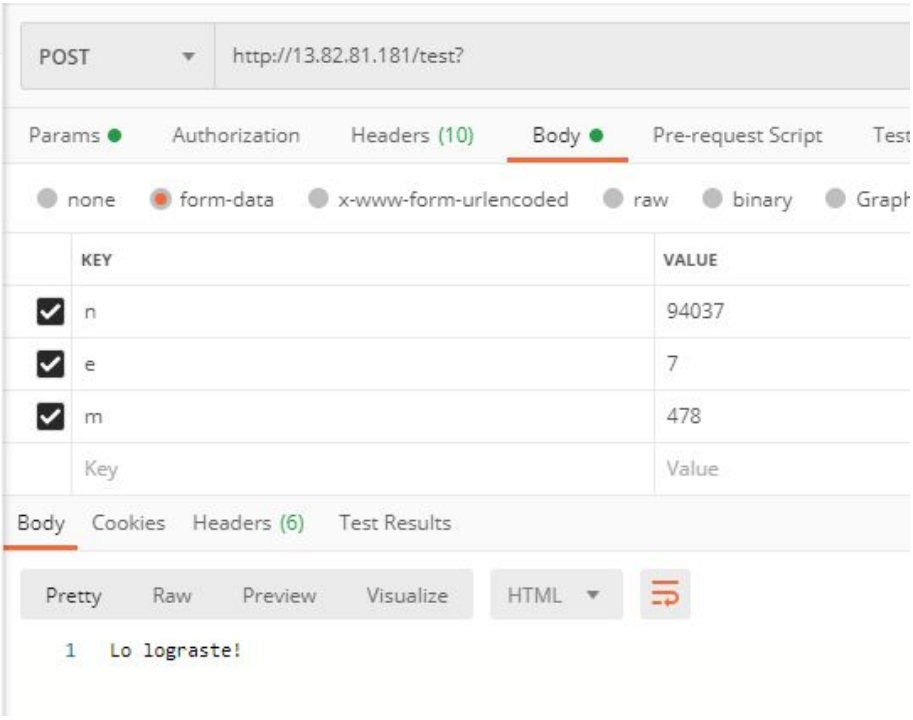
Luego se envían las llaves públicas con el programa postman, por el body y con método post a la url dada en el enunciado



Lo que nos retorna unas llaves públicas y un valor de un mensaje. Con estas llaves procedemos a cifrar el mensaje con el script de cifrar que se desarrolló anteriormente lo que nos entrega el siguiente mensaje cifrado

```
Introduzca los siguientes datos para encriptar su mensaje:
Valor de n: 899
Valor de e: 817
Valor de m: 11
Su mensaje cifrado es:  478
```

Con esta información generamos un nuevo llamado con postman, enviando nuevamente por el body los parámetros de nuestras llaves públicas y se agrega el mensaje cifrado por nuestro script.

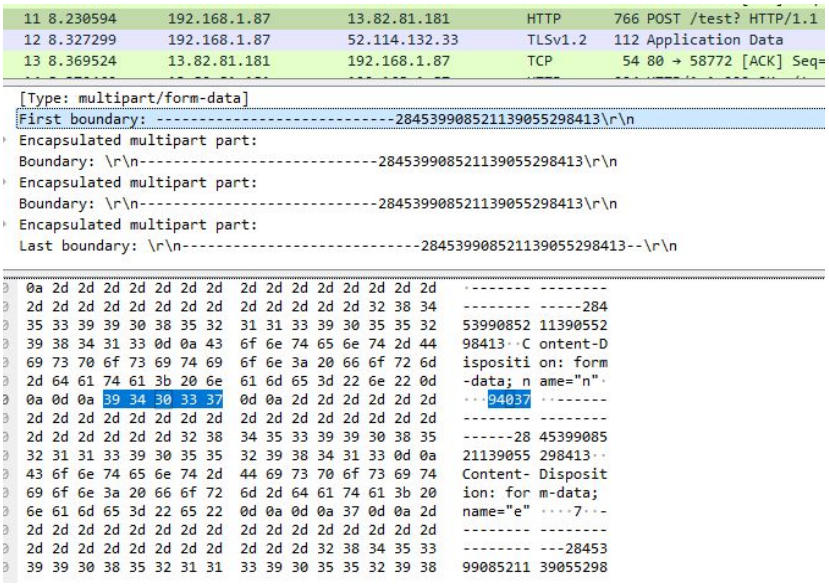


Lo que nos retorna que nuestro script cifro bien el mensaje que nos envió esta url.

Finalmente se intercepta el paquete que enviamos con postman para responder las siguientes preguntas:

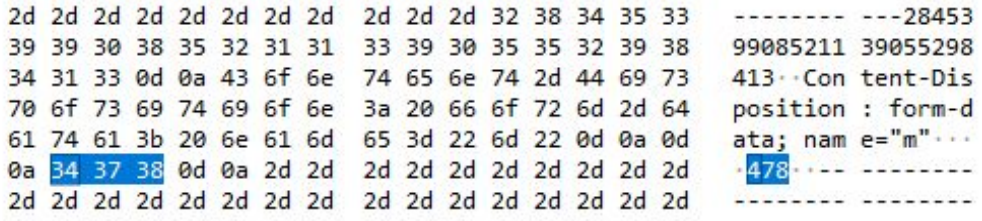
- ¿Es posible obtener información del mensaje cifrado en el segundo paso con Wireshark?
- ¿Es posible capturar la clave pública?

Lo primero que hacemos es interceptar el paquete enviado por postman hacia la url "http://13.82.81.181/test" con wireshark.



Donde podemos ver claramente las llaves públicas que enviamos por el body, y vemos que en el paquete, cada dígito de la clave la antepone un 3 y luego un dígito de nuestra llave, esto pasa para nuestro n y nuestro e,por lo que se encuentra en texto plano los parámetros que enviamos por postman.

Luego si seguimos buscando en el paquete notamos que también podemos obtener el mensaje cifrado que enviamos mediante postman y que se cifra de la misma manera que las llaves públicas que enviamos .



Conclusión:

Gracias al algoritmo de cifrado asimétrico RSA podemos compartir mensajes cifrados de forma segura, donde para que el receptor del mensaje pueda descifrar necesita solamente una de nuestras llaves públicas y el mensaje cifrado, así obtenemos un canal seguro para transmitir información importante.

Mientras más grandes sean los números primos, las llaves generadas serán más robustas ya que serán de un mayor número de dígitos, y para mayor seguridad los números escogidos no deben ser compartidos jamás, ya que con estos podemos obtener las llaves públicas y privadas que utilizamos para cifrar.