



PROGRAMA PROFESIONAL

-

TÍTULO DEL TRABAJO

Tarea - Algoritmo RSA

CURSO

Álgebra Abstracta

Alumnos:

- [Royer Diosdado Carcausto Choquehuanca](#)
- Fabricio Arián Messa Mandujano

Grupo: CCOMP3-2

Año: 2022

“El alumno declara haber realizado el presente trabajo de acuerdo a las normas de la Universidad Católica San Pablo”

1. (5 points) Si m es el mensaje y c es el cifrado (ambos representados por un entero). Y además, la clave pública es $P = \{e, n\}$ (en ese orden). Hallar m cuando:

$$P = \{65537, 999630013489\} \quad y \quad c = 747120213790$$

$$P = \{e, n\}$$

$$\text{Cifrado} = 747120213790$$

$$e = 65537$$

$$n = 999630013489$$

Según el punto 5: $d \rightarrow ed = 1 \pmod{\phi n}$ \longrightarrow Sacamos el inverso multiplicativo de "e" con ϕn

$\phi n =$	Input a number n:	Euler's totient function $\phi(n)$:	Run time:
	999630013489	Find $\phi(n)$ 999628013860	00:00:00.00E

$$e = 65537$$

Usamos el Algoritmo Extendido de Euclides y tenemos que la inversa es:

```
C:\Users\INTEL\AppData\Local\Programs\Python\Python38-64\python.exe
755383642193
Press any key to continue . . .
```

$$d = 755383642193$$

POR LO TANTO:

$$m = c^d \pmod n$$

$$m = 747120213790^{755383642193} \pmod{999630013489}$$

```
m = pow(747120213790, 755383642193, 999630013489)
print(m)
```

```
C:\Users\INTEL\AppData\Local\Programs\Python\Python38-64\python.exe
100000000001
Press any key to continue . . .
```

[Handwritten signature]

2. (7 points) Si m es el **mensaje** y c es el **cifrado** (ambos representados por un entero). Y además, la clave pública es $P = \{e, n\}$ (en ese orden). Hallar m cuando:

$$P = \{7, 35794234179725868774991807832568455403003778024228226193532908190484670252364677411513516111204504060317568667\}$$

$$c = 35794234179725868774991807832568455403003778024228226193532908190484670252364677411513516052471686245831933544$$

Sin embargo al enviar el mismo **mensaje** (m) cuando $e' = 11$, el **cifrado** resulto ser

$$c' = 35794234179725868774991807832568455403003778024228226193532908190484670252364665786748759822531352444533388184.$$

$$P = \{e, n\}$$

$$e = 7$$

$$n =$$

$$35794234179725868$$

$$77499180783256845$$

$$5403003778024228$$

$$22619353290819048$$

$$467025236467741151$$

$$3516111204504060317$$

$$568667$$

$$\text{cifrado} =$$

$$35794234179725868$$

$$77499180783256845$$

$$5403003778024228$$

$$22619353290819048$$

$$467025236467741151$$

$$35160524716862458$$

$$31933544$$

Mismo procedimiento del ejercicio 1:

$$\phi n = 1.3690042483646612e + 109$$

$$e = 7$$

Usamos nuevamente el Algoritmo Extendido de Euclides para la inversa:

```
C:\Users\iNTEL\AppData\Local\Programs\
3.911440709613318e+108
Press any key to continue . . .
```

$$d = 3.911440709613318e+108$$

Por lo tanto:

$$m = c^d \bmod n \rightarrow$$

```
28226193532908190484670252364677411513516111204504060317568667)
6193532908190484670252364677411513516111204504060317568667
Excepción producida
(34, 'Result too large')
Copiar detalles | Iniciar sesión de Live Share...
```