Criptografía y Seguridad
Tarea 2

Altamirano Vázquez Jesús Fernando
Rubí Rojas Tania Michelle

18 de mayo de 2020

1. Mediante el algoritmo de de ro-Pollar descomponer el número $n = 557437$.

   Solución: Para realizar la descomposición, haremos lo siguiente:

   a) Asignamos $a = 2$ y $b = 2$.

   b) Para $i = 1, 2, ...$
   - Se hace $a = a^2 + 1 \pmod{n}$, $b = b^2 + 1 \pmod{n}$ y $b = b^2 + 1 \pmod{n}$.
   - Calculamos $(a - b, n)$.
   - Si $(a - b, n) > 1$ se tiene que hemos encontrado un factor no trivial de n.
   - Si $(a - b, n) = 1$ o $(a - b, n) = n$ avanza a la siguiente i.

   Como $n = 557437$ entonces obtenemos lo siguiente:

i	x_i	$b = f(x_i)$	$(a - b, n)$
1	5	26	1
2	26	458,330	1
3	677	108,979	1
4	458,330	31,871	1
5	157,510	408,810	1
6	108,979	200,153	1
7	221,157	96,572	1
8	31,871	45,835	1
9	110,428	130,880	1
10	408,810	176,439	1
11	429,131	481,892	1
12	200,153	225,945	1
13	455,968	226,069	389

   Por lo tanto, la descomposición de n sería de la forma $n = (389)(1433)$.

2. Sea $\mathbb{Z}_{10007}$ y $S = \{2, 3, 5, 7\}$. Calcular el índice de $\beta = 9451$ en $\alpha = 5$.

   a) Mostrar que α es raíz primitiva de $\mathbb{Z}^*_{10007}$.
   Solución: Sabemos que un entero positivo α es una raíz primitiva módulo n si $(\alpha, n) = 1$ y $ord(\alpha) = \varphi(n)$, siendo φ la *función phi de Euler*. La *función phi de Euler* se define de la siguiente forma:

$$\varphi(m) = |\{n \in \mathbb{N} | n \leq m \wedge (m, n) = 1\}|$$

Es decir, si n es un entero positivo, entonces $\varphi(n)$ se define como el número de enteros positivos menores o iguales a n y coprimos con n.

Dada la siguiente implementación de la función Euler

```
            # Regresa el minimo comun divisor de dos numeros.
            def mcd(a, b):
                if (b == 0):
                    return a
                else:
                    return mcd(b, a % b)

            '''
            La funcion phi de Euler regresa el numero de enteros positivos que
            son menores o iguales a n y que ademas son primos relativos con n.
            '''
            def phi_Euler(n):
                relativos = []
                for i in range (1, n):
                    if (mcd(n, i) == 1):
                        relativos.append(i)
                return len(relativos)

            def main():
                print (phi_Euler(10007))

            if __name__ == "__main__":
                main()

```

obtenemos que $\varphi(10007) = 10006$ (esto es cierto ya que $n = 10007$ es un primo, y por lo tanto $\varphi(10007) = 10007 - 1 = 10006$).

El órden de un elemento x de un grupo G es el más pequeño entero positivo i tal que $x^i = e$, donde e es el elemento identidad de la multiplicación del grupo G y x^i es el producto de i veces x. Sabemos que $\mathbb{Z}^*_{10007} = \{\bar{1}, \bar{2}, \bar{3}, ..., \overline{10006}\}$. En particular, el elemento identidad con respecto a la multiplicación en $\mathbb{Z}^*_{10007}$ es $\bar{1}$ (se puede verificar con la tabla de multiplicación). Por la definición de órden, debemos hallar al más pequeño entero positivo i tal que $5^i \equiv 1$ (mód 10007).

Dada la siguiente implementación que calcula al entero i que estamos buscando y después lo compara con el órden del grupo $\mathbb{Z}^*_{10007}$ para saber si son el mismo

```
        # Nos dice si el entero "a" es una raiz primitiva de Zp.
        def es_primitiva(a, p):
            for i in range (1, p):
                if(pow(a, i, p) == 1):
                    orden = i
                    break
            return (orden == p-1) and (mcd(a, p) == 1)

        def main():
            print(es_primitiva(5, 10007))

        if __name__ == "__main__":
            main()

```

obtenemos que $i = 10006$ (para este caso en particular el resultado se puede verificar con el *Pequeño Teorema de Fermat*), y como $ord(5) = 10006 = \varphi(10007)$ y $(5, 10007) = 1$ entonces la función *es_primitiva* regresa **True**. Por lo tanto, 5 es una raíz primitiva de $\mathbb{Z}^*_{10007}$.

b) $\rho_1 = 4063$, $\rho_2 = 5136$ y $\rho_0 = 9865$. Úselos para calcular los logaritmos de 2, 3, y 7 base 5. ¿Por qué no es necesario calcular el logaritmo de 5 base α?
SOLUCIÓN: Gracias a los valores de ρ_i tenemos que

$$\begin{aligned} 5^{4063} &\pmod{10007} = 2 \times 3 \times 7 \\ 5^{5136} &\pmod{10007} = 2 \times 3^3 \\ 5^{9865} &\pmod{10007} = 3^3 \times 7 \end{aligned}$$

de donde obtenemos

$$\begin{aligned} \log_5 2 + \log_5 3 + \log_5 7 &= 4063 \pmod{10006} \\ \log_5 2 + 3\log_5 3 &= 5136 \pmod{10006} \\ 3\log_5 3 + \log_5 7 &= 9865 \pmod{10006} \end{aligned}$$

el cual es un sistema de ecuaciones con 3 incógnitas ($\log_5 2$, $\log_5 3$ y $\log_5 7$).
Resolvemos el sistema usando eliminación gaussiana:

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 1 & 3 & 0 & 5136 \\ 0 & 3 & 1 & 9865 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 2 & 10005 & 1073 \\ 0 & 3 & 1 & 9865 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 2 & 10005 & 1073 \\ 0 & 1 & 6671 & 9959 \end{array}\right) \approx$$

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 1 & 3334 & 1120 \\ 0 & 1 & 6671 & 9959 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 1 & 3334 & 1120 \\ 0 & 0 & 3337 & 8839 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 1 & 3334 & 1120 \\ 0 & 0 & 1 & 1301 \end{array}\right) \approx$$

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 4063 \\ 0 & 1 & 0 & 6190 \\ 0 & 0 & 1 & 1301 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 0 & 1 & 7879 \\ 0 & 1 & 0 & 6190 \\ 0 & 0 & 1 & 1301 \end{array}\right) \approx \left(\begin{array}{ccc|c} 1 & 0 & 0 & 6578 \\ 0 & 1 & 0 & 6190 \\ 0 & 0 & 1 & 1301 \end{array}\right)$$

donde las operaciones realizadas fueron

- $F_2 - F_1 \to F_2$
- $6671 \cdot F_3 \to F_3$ (6671 es el inverso multiplicativo de 3 en $\pmod{10006}$).
- $F_2 - F_3 \to F_2$
- $F_3 - F_2 \to F_3$
- $4003 \cdot F_3 \to F_3$ (4003 es el inverso multiplicativo de 3337 en $\pmod{10006}$).
- $F_2 - 3334 \cdot F_3 \to F_2$
- $F_1 - F_2 \to F_1$
- $F_1 - F_3 \to F_1$

Por lo anterior, obtenemos que $\log_5 2 = 6578$, $\log_5 3 = 6190$ y $\log_5 7 = 1301$.
No es necesario calcular el logaritmo de 5 base $\alpha = 5$ ya que por definición el resultado es igual a 1 ($x = log_5\ 5 \Leftrightarrow 5^x = 5 \Rightarrow x = 1$).

c) Dados los cálculos anteriores, obtener el índice de $\beta = 9451$ en α módulo 10007.
SOLUCIÓN: Tomamos $k = 7736$ de manera *aleatoria*. Calculamos

$$\beta\alpha^k = 9451 \times 5^{7736} \pmod{10007} = 8400 = 2^4 \times 3 \times 5^2 \times 7$$

Como la expresión anterior se factoriza sobre $S = \{2, 3, 5, 7\}$, obtenemos

$$\begin{aligned}
\log_5 9451 &= 4\log_5 2 + \log_5 3 + 2\log_5 5 + \log_5 7 - s \pmod{10006} \\
&= 4(6578) + 6190 + 2(1) + 1301 - 7736 \pmod{10006} \quad \text{por el inciso anterior} \\
&= 6300 + 6190 + 2 + 1301 - 7736 \pmod{10006} \\
&= 6057
\end{aligned}$$

Así, el índice que buscamos es 6057 (para verificarlo podemos comprobar que efectivamente $5^{6057} \equiv 9451 \pmod{10007}$).

3. Descifrar el siguiente mensaje en RSA con parámetros $(2257, 7)$ con las siguientes condiciones:

   a) Aplicar el algoritmo de Criba Cuadrática para descomponer a 2257.
   - Usar la base $S = \{-1, 2, 3, 17\}$.
   - Dar M, B y decir para qué sirven.
   - Explicar el proceso por el cual se obtienen x e y, con los cuales se puede descomponer 2257.
   - Dar los valores para los cuales se obtienen x e y tales que $(x - y, 2257)$ es un factor no trivial de 2257.

   SOLUCIÓN: Primero calculamos los parámetros M y B que determinarán el tamaño del intervalo de la criba y de la base de los factores, respectivamente.

$$M = \lfloor (e^{\sqrt{\ln(2257)\ \ln(\ln(2257))}})^{\frac{3\sqrt{2}}{4}} \rfloor = 67$$

$$B = \lfloor (e^{\sqrt{\ln(2257)\ \ln(\ln(2257))}})^{\frac{\sqrt{2}}{4}} \rfloor = 4$$

   De este modo ya tenemos el intervalo de la criba $[-67, 67]$. Además, podemos verificar que efectivamente la cardinalidad de la base de factores $S = \{-1, 2, 3, 17\}$ es igual a 4.
   Luego, se buscan parejas (a_i, b_i) con $a_i^2 \equiv b_i \pmod{n}$. Para obtener estas parejas, haremos lo siguiente:
   Calculamos

$$m = \lfloor \sqrt{n} \rfloor = \lfloor \sqrt{2257} \rfloor = 47$$

   y formamos el polinomio

$$q(x) = (x + m)^2 - n = (x + 47)^2 - 2257$$

   Tenemos que $a_i = x + 47$ ya que

$$\begin{aligned}
a^2 \equiv b_i &= q(i) \\
&= (i + 47) - 2257, \quad i \in \{0, 1, -1, 2, -2, 3, -3, ...\}
\end{aligned}$$

   Debemos expresar $b_i = \prod_{k=1}^{s} p_k^{e_{ij}}$. Esto lo podemos observar en la siguiente tabla:

x	$b_i = q(x)$	$b_i = \prod_{i=1}^{4} p_i$	a_i	w_i	v_i
0	-48	$(-1^1)(2^4)(3^1)(17^0)$	47	$(1,4,1,0)$	$(1,0,1,0)$
1		No se puede expresar en términos de la base			
-1		No se puede expresar en términos de la base			
2	144	$(-1^0)(2^4)(3^2)(17^0)$	49	$(0,4,2,0)$	$(0,0,0,0)$
-2		No se puede expresar en términos de la base			
3	243	$(-1^0)(2^0)(3^5)(17^0)$	50	$(0,0,5,0)$	$(0,0,1,0)$
-3		No se puede expresar en términos de la base			
4		No se puede expresar en términos de la base			
-4	-408	$(-1^1)(2^3)(3^1)(17^1)$	43	$(1,3,1,1)$	$(1,1,1,1)$
5		No se puede expresar en términos de la base			
-5		No se puede expresar en términos de la base			
6		No se puede expresar en términos de la base			
-6	-576	$(-1^1)(2^6)(3^2)(17^0)$	41	$(1,6,2,0)$	$(1,0,0,0)$

La tabla para fines prácticos la podemos ver de la siguiente forma:

i	x	$b_i = q(x)$	$b_i = \prod_{i=1}^{4} p_i$	a_i	w_i	v_i
1	0	-48	$(-1^1)(2^4)(3^1)(17^0)$	47	$(1,4,1,0)$	$(1,0,1,0)$
2	2	144	$(-1^0)(2^4)(3^2)(17^0)$	49	$(0,4,2,0)$	$(0,0,0,0)$
3	3	243	$(-1^0)(2^0)(3^5)(17^0)$	50	$(0,0,5,0)$	$(0,0,1,0)$
4	-4	-408	$(-1^1)(2^3)(3^1)(17^1)$	43	$(1,3,1,1)$	$(1,1,1,1)$
5	-6	-576	$(-1^1)(2^6)(3^2)(17^0)$	41	$(1,6,2,0)$	$(1,0,0,0)$

En particular tenemos que $v_1 + v_3 + v_5 = 0$. Por lo que, $T = \{1, 3, 5\}$.
Procedemos a calcular

$$\begin{aligned} x &= a_1 a_3 a_5 \pmod{2257} \\ &= 47 \cdot 50 \cdot 41 \pmod{2257} \\ &= 1556 \end{aligned}$$

y los valores de l_j (los cuales obtenemos al sumar coordenada a coordenada los vectores w_i con $i \in T$, y el valor que obtenemos después de esta suma lo dividimos entre 2)

$$l_1 = 1, l_2 = 5, l_3 = 4, l_4 = 0$$

gracias a los cuales podemos calcular

$$y = 1^1 \cdot 2^5 \cdot 3^4 \cdot 7^0 = -2^5 \cdot 3^4 = 2592$$

Como $1556 \not\equiv \pm 2592 \pmod{2257}$ entonces sólo queda obtener

$$(x - y, n) = (1556 - 2592, 2257) = (-1036, 2257) = 37$$

y

$$(x + y, n) = (1556 + 2592, 2257) = (4148, 2257) = 61$$

Por lo tanto, 2257 tiene dos factores no triviales que son 37 y 61.

b) Descifrar el mensaje mediante el siguiente proceso:

- Dar las llaves pública y privada de RSA y su proceso de cómo se obtienen de manera resumida pero clara.
  SOLUCIÓN: Sabemos que los parámetros del mensaje RSA son $(2257, 7)$, por lo que $p = 37$ y $q = 61$ (por el inciso anterior).
  Entonces $n = pq = 37 \cdot 61 = 2257$, y obtenemos

$$\begin{aligned}
\varphi(n) &= \varphi(37 \cdot 61) \\
&= \varphi(37)\varphi(61) \\
&= (37-1)(61-1) && \text{ya que } p \text{ y } q \text{ son primos} \\
&= 36 \cdot 60 \\
&= 2160
\end{aligned}$$

  Después, elegimos una llave de cifrado e, que es un entero impar (pues $(p-1)$ y $(q-1)$ son pares) tal que $0 < e < \varphi(n)$ y es primo relativo con $\varphi(n)$. En este caso, por los parámetros del mensaje RSA tenemos que $e = 7$ (el cual, efectivamente cumple las propiedades descritas anteriormente).
  Ahora, debemos calcular el número d tal que $0 \leq d \leq n$ y que cumple

$$ed \equiv 1 \pmod{\varphi(n)}$$

  Esto lo podemos obtener usando el algoritmo extendido de Euclides, ya que d es el inverso multiplicativo de e módulo $\varphi(n)$.
  Dada la siguiente implementación del algoritmo extendido de Euclides

```
'''
Regresa una tupla (mcd, s, t) que obtenemos al aplicar el
algoritmo extendido de Euclides, donde as + bt = mcd(a, b)
son los elementos que conforman la tupla.
'''
def aee(a, b):
    s = 0; s_i = 1
    t = 1; t_i = 0
    g = b; g_i = a

    while g != 0:
        cociente = g_i // g
        g_i, g = g, g_i - cociente * g
        s_i, s = s, s_i - cociente * s
        t_i, t = t, t_i - cociente * t
    return (g_i, s_i, t_i)

# Calcula el inverso de 7 en Z_{2160}.
def main():
    g, s, t = aee(7, 2160)
    # El inverso multiplicativo de a modulo m existe si y
    # solo si (a,m) = 1.
    if g != 1:
        print("No tiene inverso multiplicativo.")
    else:
        inverso = s % 2160
        print(inverso)

if __name__ == "__main__":
    main()

```

obtenemos que $d = 1543$.
Por lo tanto, la llave pública es la tupla

$$P = (e, n) = (7, 2160)$$

y la llave privada, que se mantiene secreta, es $d = 1543$, o bien, la tupla

$$S = (d, n) = (1543, 2160)$$

- Descifrar el mensaje.

```
585 1660 585 2011 431 322 431 322 274 585 322 431 585 1660
68 322 1660 1933 1132 128 1995 322 2218 322 128 399 585 1660
128 399 322 585 2011 1933 1132 1411 2011 585 1660 128 399 233
233 322 2218 585 274 319 2011 585 1660 128 399 233 233 319
1660 319 2011 399 68 1660 399 1387 399 128 322 274 322 2218
399 2187 319 2011 399 68 1660 399 1387 399 128 322 585 1660
585 2011 431 585 128 322 2011 319 1418 1132 585 2011 322 233
585 128 319 1660 233 319 1 322 2011 399 128 319 1411 322 284
585 274 322 1660 1418 1132 585 585 2011 431 319 585 2011 431
322 1933 1132 1411 2187 399 284 585 274 431 399 2187 319
```

SOLUCIÓN: Para recuperar el mensaje original a partir del cifrado se realiza la siguiente operación:

$$M = c^d \pmod{2257}$$

donde c es el texto cifrado y d es nuestra llave privada.
Dada la siguiente implementación

```python
# Descifra un mensaje utilizando RSA.
def descifrar(d, n):
    # Leemos el archvio de texto donde se encuentra el mensaje.
    file = open('texto.txt', 'r')
    data = file.readlines()
    file.close()

    msg = []
    # Calculamos c^d (mod n) para cada uno de los elementos
    # del mensaje.
    for renglon in data:
        for palabra in renglon.split(' '):
            msg.append(pow(int(palabra), d, n))
    print(msg)

    modulo = []
    # Le aplicamos (mod 26) a cada uno de los elementos del
    # mensaje descifrado.
    for elemento in msg:
        modulo.append(elemento % 26)
    print(modulo)

def main():
    descifrar(1543, 2257)

if __name__ == "__main__":
    main()

```

tenemos que la función *descifrar* nos regresa dos listas: la primera es el resultado de aplicar la operación $c^d \pmod{n}$ a cada uno de los elementos del mensaje, mientras que la segunda utiliza esta primer lista y le aplica módulo 26 (ya que los elementos obtenidos al descifrar el mensaje están en un rango de $[1, 26]$ y al momento de sustituir queremos que los elementos en un rango de $[0, 25]$).
La primer lista se ve como

```
[4, 13, 4, 18, 19, 26, 19, 26, 17, 4, 26, 19, 4, 13,
6, 26, 13, 12, 20, 2, 7, 26, 15, 26, 2, 8, 4, 13, 2,
8, 26, 4, 18, 12, 20, 24, 18, 4, 13, 2, 8, 11, 11,
26, 15, 4, 17, 14, 18, 4, 13, 2, 8, 11, 11, 14, 13,
14, 18, 8, 6, 13, 8, 5, 8, 2, 26, 17, 26, 15, 8, 3,
14, 18, 8, 6, 13, 8, 5, 8, 2, 26, 4, 13, 4, 18, 19,
4, 2, 26, 18, 14, 16, 20, 4, 18, 26, 11, 4, 2, 14,
13, 11, 14, 1, 26, 18, 8, 2, 14, 24, 26, 21, 4, 17,
26, 13, 16, 20, 4, 4, 18, 19, 14, 4, 18, 19, 26, 12,
20, 24, 3, 8, 21, 4, 17, 19, 8, 3, 14]

```

mientras que la segunda lista es de la forma

```
[4, 13, 4, 18, 19, 0, 19, 0, 17, 4, 0, 19, 4, 13, 6,
0, 13, 12, 20, 2, 7, 0, 15, 0, 2, 8, 4, 13, 2, 8, 0,
4, 18, 12, 20, 24, 18, 4, 13, 2, 8, 11, 11, 0, 15, 4,
17, 14, 18, 4, 13, 2, 8, 11, 11, 14, 13, 14, 18, 8,
6, 13, 8, 5, 8, 2, 0, 17, 0, 15, 8, 3, 14, 18, 8, 6,
13, 8, 5, 8, 2, 0, 4, 13, 4, 18, 19, 4, 2, 0, 18, 14,
16, 20, 4, 18, 0, 11, 4, 2, 14, 13, 11, 14, 1, 0, 18,
8, 2, 14, 24, 0, 21, 4, 17, 0, 13, 16, 20, 4, 4, 18,
19, 14, 4, 18, 19, 0, 12, 20, 24, 3, 8, 21, 4, 17, 19,
8, 3, 14]

```

Finalmente, usaremos esta segunda lista para obtener el mensaje original.
Dada la siguiente tabla, en donde asignamos a cada letra un número

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

podemos concluir que el mensaje es:

```
EN ESTA TAREA TENGAN MUCHA PACIENCIA ES MUY SENCILLA PERO
SENCILLO NO SIGNIFICA RAPIDO SIGNIFICA EN ESTE CASO QUE
SALE CON LO BASICO YA VERAN QUE ESTO ESTA MUY DIVERTIDO.
```