PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Criptografía y Seguridad Computacional - IIC3253
Ayudantía 7
Alexander Pinto

RSA

1. Realice una demostración práctica del protocolo RSA, para ello considere $P = 29$, $Q = 37$ y $m = 125$.

   Siguiendo los pasos del protocolo criptográfico RSA de la siguiente forma:

   1. Dos números primos $P = 29$ y $Q = 37$. Definimos N:

   $$N = P \cdot Q$$
   $$N = 29 \cdot 37$$
   $$N = 1073$$

   2. Definimos $\phi(N)$:

   $$\phi(N) = (P-1) \cdot (Q-1)$$
   $$\phi(N) = 28 \cdot 36$$
   $$\phi(N) = 1008$$

   3. Generamos un número d tal que $MCD(d, \phi(N)) = 1$.

   Para ello probamos con distintos d y podemos comprobar que $MCD(d, \phi(N)) = 1$ con el uso del algoritmo de Euclides extendido y la identidad de Bezout, $MCD(a, b) = s \cdot a + t \cdot b$. Detalles de la ejecución de este algoritmo se encuentran en la Ayudantía 7.

   Probamos para $d = 515$, tomando para el algoritmo $a = d$ y $b = \phi(N)$. En resumen, tenemos para el cálculo del resto r_i:

$$
\begin{aligned}
r_{i+1} = s_i \quad & \times a \quad + \quad t_i \times b \\
r_0 = 1 \quad & \times 1008 \quad + \quad 0 \times 515 = 1008 \\
r_1 = 0 \quad & \times 1008 \quad + \quad 1 \times 515 = 515 \\
r_2 = 1 \quad & \times 1008 \quad + \quad -1 \times 515 = 493 \\
r_3 = -1 \quad & \times 1008 \quad + \quad 2 \times 515 = 22 \\
r_4 = 23 \quad & \times 1008 \quad + \quad -45 \times 515 = 9 \\
r_5 = -47 & \times 1008 \quad + \quad 92 \times 515 = 4 \\
r_6 = 117 & \times 1008 \quad + -229 \times 515 = 1
\end{aligned}
$$

Con esto comprobamos que d cumple con este criterio.

4. Construya un número e tal que $e \cdot d \equiv 1 \mod \phi(N)$.

El algoritmo de Euclides extendido nos proporciona también el número e:

$$
\begin{aligned}
e &= t \mod \phi(N) \\
e &= -229 \mod 1008 \\
e &= 779
\end{aligned}
$$

5. Defina $S_A = (d, N)$ y $P_A = (e, N)$

$$
\begin{aligned}
S_A &= (515, 1073) \\
P_A &= (779, 1073)
\end{aligned}
$$

6. Cifrado:

$$
\begin{aligned}
Enc_{P_A} &= m^e \mod N \\
Enc_{P_A} &= 125^{779} \mod 1073 \\
c &= 267
\end{aligned}
$$

7. Descifrado:

$$
\begin{aligned}
Dec_{S_A} &= c^d \mod N \\
Dec_{S_A} &= 267^{515} \mod 1073 \\
m &= 125
\end{aligned}
$$

2. Explique Realice una demostración práctica del protocolo RSA, para ello considere:

$P = 113863597804193638684746233303866275862959799351116319454932839076728438946896147861014045230382038476640030287985809455541219549331656514489547597100776866463781814271673557022224794131215064237956990930271588711457443015898310055178700526009643888726738585998120562566732806649886086397912653034050178893859$

$Q = 11583890979194374892298775385105033456019334360883894316746576272187248423890728330560926045219543313224793521803302492498157940505127336037175398388575365495723472206444658775444594608062194308816049757895615186706312698865232167216144843968855048890437216703536662605184117506459074674348381243431062066279 3$

$m = 123456789987654321.$

¿Qué inconvenientes espera encontrar en su demostración?

- Primero necesitamos comprobar que P y Q son realmente primos.
- Requerimos una función que nos permite hallar d de manera eficiente.
- Necesitamos hallar la potencia de números grandes. Felizmente contamos con la función `pow` de Python!
- Finalmente necesitamos implementar el protocolo RSA.

3. Una aplicación práctica de la criptografía asimétrica son las firmas digitales. La idea de una firma digital es dar autenticidad a un mensaje o documento. Para ello, un usuario A utiliza su clave privada para "firmar" un documento m, y cualquier otro usuario puede verificar que este documento realmente fue firmado por A utilizando la clave pública de A. En particular, en el protocolo de criptografía RSA se cumple la propiedad $m = Enc_{P_A}(Dec_{S_A}(m))$, útil para esta aplicación. Demuestre esta propiedad.

   Considerando que:

$$Enc_{P_A}(m) = m^e \mod N$$
$$Dec_{S_A}(c) = c^d \mod N$$

   tenemos que:

$$m = Enc_{P_A}(m^d \mod N)$$
$$m = (m^d \mod N)^e \mod N$$
$$m = m^{d \cdot e} \mod N$$

   Dado que $m \in \{0, ..., N-1\}$, esto es equivalente a:

$$m^{d \cdot e} \equiv m \mod N$$

   Consideramos los siguientes casos:

   (1) $MCD(m, N) = 1$.

   Dado esto tenemos que $MCD(m, P) = 1$, luego por el pequeño teorema de Fermat:

$$m^{P-1} \equiv 1 \mod P$$

, tenemos:

$$\begin{aligned}\left(m^{P-1}\right)^{Q-1} &\equiv 1 \mod P\\ m^{\phi(N)} &\equiv 1 \mod P\\ m^{\alpha\cdot\phi(N)} &\equiv 1 \mod P\\ m^{\alpha\cdot\phi(N)+1} &\equiv m \mod P\end{aligned}$$

por lo tanto:

$$m^{e\cdot d} \equiv m \mod P$$

De la misma forma concluimos que:

$$m^{e\cdot d} \equiv m \mod Q$$

Ambas congruencias, podemos expresarlas como:

$$\begin{aligned}m^{e\cdot d} - m &= \beta \cdot P\\ m^{e\cdot d} - m &= \gamma \cdot Q\end{aligned}$$

Como $\beta \cdot P = \gamma \cdot Q$, y dado que P y Q son primos, P divide a γ, entonces, $\gamma = \delta \cdot P$, y como $m^{e\cdot d} - m = \gamma \cdot Q$, concluimos que $m^{e\cdot d} - m = \delta \cdot P \cdot Q$, es decir, $m^{e\cdot d} - m = \delta \cdot N$, para finalmente concluir que:

$$m^{e\cdot d} \equiv m \mod N$$

(2) $MCD(m, N) > 1$.

Si $m = 0$ se concluye trivialmente que $m^{e\cdot d} \equiv m \mod N$.

Luego, tenemos los siguientes casos:

• Si P divide a m, pero Q no divide a m, entonces:

$$\begin{aligned}m^{e\cdot d} &\equiv m \mod P\\ m &\equiv 0 \mod P\\ m^{e\cdot d} &\equiv m \mod Q\end{aligned}$$

dado que $MCD(m, Q) = 1$. Luego, concluimos de la misma manera que para la parte (1):

$$m^{e\cdot d} \equiv m \mod N$$

• Si Q divide a m, pero P no divide a m, seguimos el mismo razonamiento que el caso anterior y concluimos:

$$m^{e\cdot d} \equiv m \mod N$$

Finalmente de (1) y (2), hemos demostrado que en el protocolo de criptografía RSA se cumple la propiedad $m = Enc_{P_A}(Dec_{S_A}(m))$.

4. Investigue acerca del concepto de cifrado homomórfico. Luego demuestre que el protocolo de criptografía RSA es homomórfica en la multiplicación.

   Se dice que un sistema de cifrado es homomórfico si tiene la propiedad que al realizar operaciones sobre datos cifrados, y posteriormente se descifra el resultado, se obtendrá el mismo resultado si se hubiera realizado operaciones equivalentes sobre los datos originales.

   Demostraremos entonces que en el protocolo de criptografía RSA se cumple que:

$$m \times n = Dec_{S_A}(Enc_{P_A}(m) \times Enc_{P_A}(n))$$

   Considerando que:

$$Enc_{P_A}(m) = m^e \mod N$$
$$Dec_{S_A}(c) = c^d \mod N$$

   tenemos que:

$$\begin{aligned} m \cdot n &= Dec_{S_A}(m^e \mod N \times n^e \mod N) \\ m \cdot n &= (m^e \mod N \times n^e \mod N)^d \mod N \\ m \cdot n &= ((m^e \cdot n^e) \mod N)^d \mod N \\ m \cdot n &= (m^e \cdot n^e)^d \mod N \\ m \cdot n &= (m \cdot n)^{e \cdot d} \mod N \end{aligned}$$

   Luego si hacemos $p = m \cdot n$, tenemos:

$$p = p^{e \cdot d} \mod N$$

   equivalente a:

$$p^{e \cdot d} \equiv p \mod N$$

   y podemos continuar nuestra demostración siguiendo los dos casos de la pregunta anterior.