Facultad de Ciencias de la Universidad Nacional Autónoma de México

Tarea 2 de Criptografía y Seguridad

Parcial 1

Oscar Andrés Rosas Hernandez

Jorge Luís García De Santiago

29 de octubre de 2019

Índice general

Capítulo 1

Símbolo de Legendre y sus propiedades

2.1 Símbolo de Legendre y sus propiedades.

Sea p un primo impar. Diremos que un número natural a primo con p es un *resto cuadrático* módulo p si $x^2 \equiv a(mód.p)$, para cierto entero x. En caso contrario, siempre suponiendo que p es primo, diremos que a es un *resto no cuadrático*. Se conoce como *símbolo de Legendre* a la expresión

$$\left(\frac{a}{p}\right) = (a/p) = \begin{cases} 1 \text{ si } a \text{ es resto cuadratico } módulo\ p \\ -1 \text{ si } a \text{ es resto no cuadrático } módulo\ p \\ 0 \text{ si } p \mid a \end{cases}$$

Si a es un resto cuadrático respecto al módulo p, se tiene $a^{\frac{p-1}{2}} \equiv 1(mód.p)$. Si a es un no cuadrático respecto al módulo p, se tiene $a^{\frac{p-1}{2}} \equiv -1(mód.p)$. En efecto, según el teorema de Fermat, $a^{p-1} \equiv 1(mód.p)$ donde $\left(a^{\frac{p-1}{2}} - 1\right)\left(a^{\frac{p-1}{2}} + 1\right) \equiv 0(mód.p)$. De aquí podemos deducir que $a^{(p-1)/2} \equiv \left(\frac{a}{p}\right)(mód.p)$ que nos permite la solución a la ecuación $x^2 \equiv a(mód.p)$ aplicando propiedades del símbolo de Legendre.
El símbolo de Legendre satisface algunas propiedades interesantes como:

i. Si $a \equiv b(mód.p)$, entonces $\left(\frac{a}{p}\right) = \left(\frac{b}{p}\right)$. Esta propiedad se debe a que los números de una misma clase son simultáneamente restos o no restos cuadráticos.

ii. Si $a \equiv 1(mód.p)$, tenemos $\left(\frac{1}{p}\right) = 1$. En efecto, $1 = 1^2$ y, por tanto, 1 es un resto cuadrático.

iii. Si $a \equiv -1(mód.p)$, entonces $\left(\frac{-1}{p}\right) = (-1)^{\frac{p-1}{2}}$. Esta propiedad se deduce de la anterior para $a = -1$ y denota un resto no cuadrático.

iv. Si $a \equiv 2(mód.p)$, entonces $\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}$.

v. Si $a \equiv -3(mód.p)$, entonces $\left(\frac{-3}{p}\right) = \begin{cases} 1 \text{ si } p \equiv 1(mód.6) \\ -1 \text{ si } p \equiv 5(mód.6) \end{cases}$.

vi. Si $a \equiv 5(mód.p)$, entonces $\left(\frac{5}{p}\right) = \begin{cases} 1 \text{ si } p \equiv 1,9(mód.10) \\ -1 \text{ si } p \equiv 3,7(mód.10) \end{cases}$.

vii. Sea $\left(\frac{abb}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)\left(\frac{b}{p}\right)$. Se deduce, en particular, que $\left(\frac{ab^2}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{a}{p}\right)$, ya que $\left(\frac{b}{p}\right)^2 = 1$. Esto significa que en el numerador del *símbolo de Legendre* se puede eliminar cualquier factor cuadrático.

viii. Si p y q son números primos impares, $\left(\frac{q}{p}\right) = (-1)^{\left(\frac{p-1}{2}\right)\left(\frac{q-1}{2}\right)}\left(\frac{p}{q}\right)$. Esta propiedad es conocida como *Ley de Reciprocidad Cuadrática*.

Nota: La *Ley de Reciprocidad Cuadrática* tiene un papel muy importante en la teoría de los números, ya que en base a ésta, se han obtenido otros resultados interesantes en diversos campos de las matemáticas. Descubierta por $Euler$ ($1707-1783$) en 1742, gracias a los trabajos realizados por $Fermat$ $(1601-1665,$ revisada en 1772, fue publicada en su *Opuscula Analitica* de 1873, después de su muerte. $Legendre$ $(1752-1833)$ fue otro de los pioneros en el estudio de esta ley, de hecho fue el primero en dar una demostración. Basándose en los trabajos de Euler, en 1798 publica en su obra *Essai sur la Théorie des Nombres* un lema que hoy se conoce como *símbolo de Legendre*. El primero que ofrece una demostración completa de la *Ley de Reciprocidad Cuadrátic*a fue $Gauss$ $(1777-1855)$, a la que llama *Theorema Aureum* (Teorema áureo), recogida en su obra Disquisitiones Arithmeticae y publicada en 1796.

2.2 Símbolo de Jacobi y sus propiedades.

Consideremos el símbolo $\left(\frac{n}{m}\right)$ ó (n/m) para números impares m con $m>1,$ no necesariamente primos, donde $m=p_1\cdot p_2\cdot\ldots\cdot p_n,$ con $mcd(n,m)\neq 1.$
El *Símbolo de Jacobi* se define como

$$\left(\frac{n}{m}\right)=\left(\frac{a}{p_1}\right)\left(\frac{b}{p_2}\right)\ldots\left(\frac{l}{p_n}\right)$$

Sus propiedades son similares a las propiedades al S*ímbolo de Legendre*.
El uso del S*ímbolo de Jacobi* proporciona la generalización del S*ímbolo de Legendre* y la del teorema de los recíprocos cuadráticos $\left(\frac{m}{n}\right)\left(\frac{n}{m}\right)=(-1)^{(m-1)(n-1)/4}$, para m,n relativamente primos enteros, con $n\geq 3.$ Esta igual es equivalente a

$$\left(\frac{m}{n}\right)=(-1)^{(m-1)(n-1)/4}\left(\frac{n}{m}\right)$$

que también podemos escribir como:

$$\left(\frac{n}{m}\right)=\begin{cases}+\left(\frac{m}{n}\right) \text{ para } m \text{ ó } n\equiv 1(mód.4)\\ -\left(\frac{m}{n}\right) \text{ para } \quad m,n\equiv 3(mód.4)\end{cases}.$$

Estos es lo que hemos definido anteriormente como *Ley de Reciprocidad Cuadrática*.

2.3 Demostrar que si p es primo y a y b son dos enteros con $a\equiv b(mód.p),$ entonces $\left(\frac{a}{p}\right)=\left(\frac{b}{p}\right).$

El valor de $\left(\frac{a}{p}\right)$ depende sólo de si a es restos cuadrático, esto es, si $x^2\equiv a(mód.p)$ tiene solución. Como esto sólo depende de la clase de equivalencia de a respecto a $p,$ se verifica que $\left(\frac{a}{p}\right)=\left(\frac{b}{p}\right)$ si, y sólo si $a\equiv b(mód.p).$

Si p es primo y $mcd(a,p)=1,$ obtenemos $\left(\frac{a}{p}\right)\equiv a^{(p-1)/2}(mód.p),$ que es el *criterio de Euler*.

Por el *pequeño teorema de Fermat* sabemos que $a^{p-1}\equiv 1(mód.p),$ esto nos permite deducir que $\left(a^{p-1}\right)^2\equiv a^{p-1}\equiv\pm 1(mód.p),$ y para que $a^{p-1}\equiv 1(mód.p)$ será necesario que $\left(\frac{a}{p}\right)=1.$

Por ejemplo, si $a=7$ y p es de la forma $p\equiv 1(mód.4),$ tenemos

$$\left(\frac{7}{p}\right)=\left(\frac{p}{7}\right)=\begin{cases} 1\ si\ p\equiv 1,2,4(mód.7)\\ -1\ si\ p\equiv 3,5,6(mód.7)\end{cases}$$

Si p es de la forma $p\equiv 3(mód.4),$ tenemos

$$\left(\frac{7}{p}\right)=-\left(\frac{p}{7}\right)=\begin{cases} 1\ si\ p\equiv 3,5,6(mód.7)\\ -1\ si\ p\equiv 1,2,4(mód.7)\end{cases}$$

o, también

$$\left(\frac{7}{p}\right)=\begin{cases} 1\ si\ p\equiv 1,3,9,19,25,27(mód.28)\\ -1\ si\ p\equiv 5,11,13,15,17,23(mód.28)\end{cases}$$

2.4 Demostrar que si p es primo y a y b son dos enteros no divisibles con $q,$ entonces $\left(\frac{ab}{p}\right)=\left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$

Sabemos que $a^{(p-1)/2}\equiv\left(\frac{a}{p}\right)(mód.p),$ entonces $(ab)^{(p-1)/2}\equiv\left(\frac{ab}{p}\right)(mód.p)$ donde $mcd(ab,p)=1,$

como $a^{(p-1)/2}\equiv\left(\frac{a}{p}\right)(mód.p)$ y $b^{(p-1)/2}\equiv\left(\frac{b}{p}\right)(mód.p),$ se cumple que $\left(\frac{ab}{p}\right)\equiv\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)(mód.p),$ y

como p no es par, $\left(\frac{ab}{p}\right)=\left(\frac{a}{p}\right)\left(\frac{b}{p}\right),$ luego $\left(\frac{ab}{p}\right)=a^{(p-1)/2}b^{(p-1)/2}=(ab)^{(p-1)/2}\equiv\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)(mód.p).$

2.5 Demostrar que si p es impar, entonces $\left(\frac{2}{p}\right)=(-1)(p^2-1)/2.$

Sea $\left(\frac{a}{p}\right)=(-1)^k$ donde k es el número de restos que son mayores que $p/2,$ como $(-1)^8\equiv k(mód.2),$ entonces

$$\left(\frac{2}{p}\right)=(-1)^k=(-1)^{(p^2-1)/8}$$

El profesor Vinogradov llega a esta conclusión utilizando el S*ímbolo de Jacobi*. Este matemático dice que si *P* es impar mayor que la unidad, esto es, $m = p_1 \cdot p_2 \cdot \ldots \cdot p_n$ que es la descomposición factorial de P, y si el $mcd(a,P)=1$, entonces el Símbolo de Jacobi $\left(\frac{a}{m}\right)$ se define por la igualdad:

$$\left(\frac{a}{m}\right)=\left(\frac{a}{p_1}\right)\cdot\left(\frac{a}{p_2}\right)\cdot\ldots\cdot\left(\frac{a}{p_r}\right)=\left(\frac{2}{p}\right)=(-1)^{(p^2-1)/8}$$

A partir de lo expuestos podemos obtener otras igualdades como

$$\left(\frac{a}{p}\right)=\left(\frac{ab^2}{p}\right),\ \left(\frac{1}{p}\right)=1 \text{ ó } \left(\frac{1}{p}\right)=(-1)^{(p-1)/2}\ \left(\frac{1}{p}\right)=(-1)^{(p-1)/2}$$

y también

$$\left(\frac{2a}{p}\right)=\left(\frac{2a+2p}{p}\right)=\left(\frac{4\dfrac{a+p}{2}}{p}\right)=\left(\frac{\dfrac{a+p}{2}}{p}\right)$$

que nos lleva a que

$$\left(\frac{2}{p}\right)\left(\frac{a}{p}\right)=(-1)^{\sum_{x=1}^{\frac{p}{2}}\left[\frac{ax}{p}\right]+\frac{p^2-1}{8}},$$

de la que nos permite deducir dos propiedades muy importantes del *Símbolo de Legendre*. La primera es que para $a=1$:

$$\left(\frac{2}{p}\right)=(-1)^{(p^2-1)/8}$$

la segunda es que si *p* es de la forma $p=8m+s,$ donde *s* es uno de los números *1, 3, 5, 7*, y además

$$\left(\frac{p^2-1}{8}\right)=8m^2+2ms+\frac{s^2-1}{8}$$

entonces este número será par si $s=1$ ó $s=7,$ e impar si $s=3$ ó $s=5.$ Por tanto, el número *2* será resto cuadrático respecto al módulo *p* si *p* es de la forma $p=8m+1$ ó $p=8m+7$ y será no resto cuadrático respecto al módulo *p* si *p* es de la forma $p=8m+3$ ó $p=8m+5.$

2.6 Demostrar que si P y Q son números impares positivos, primos entre sí, entonces $\left(\frac{Q}{P}\right)=(-1)^{(P-1)(Q-1)/4}\left(\frac{P}{Q}\right)$.

Como $\left(\frac{P-1}{2}\right)\cdot\left(\frac{Q-1}{2}\right)$ es impar solamente cuando ambos números, P y Q, son de la forma $4m+3$, y es par si al menos uno de estos números es de la forma $4m+1$, la propiedad señalada se puede formular así:

Si ambos números, P y Q, son de la forma $4m+3$, entonces $\left(\frac{Q}{P}\right)=-\left(\frac{P}{Q}\right)$.

Si al menos uno de los números, P y Q, es de la forma $4m+1$, entonces $\left(\frac{Q}{P}\right)=\left(\frac{P}{Q}\right)$.

Supongamos que $Q=q_1\cdot q_2\cdot\ldots\cdot q_n$ es la descomposición de Q en factores primos, se tiene

$$\left(\frac{Q}{P}\right)=\left(\frac{Q}{p_1}\right)\cdot\left(\frac{Q}{p_2}\right)\cdot\ldots\cdot\left(\frac{Q}{p_n}\right)=\prod_{\alpha=1}^{s}\cdot\prod_{\beta=1}^{t}\left(\frac{Q_\beta}{P_\alpha}\right)$$

y como

$$\prod_{\alpha=1}^{s}\cdot\prod_{\beta=1}^{t}\left(\frac{Q_\beta}{P_\alpha}\right)=(-1)^{\left(\sum_{\alpha=1}^{r}\frac{P_\alpha-1}{2}\right)\left(\sum_{\beta=1}^{s}\frac{Q_\beta-1}{2}\right)}\prod_{\alpha=1}^{s}\cdot\prod_{\beta=1}^{t}\left(\frac{P_\alpha}{Q_\beta}\right)$$

resulta:

$$\left(\frac{Q}{P}\right)=(-1)^{\left(\sum_{\alpha=1}^{r}\frac{P_\alpha-1}{2}\right)\left(\sum_{\beta=1}^{s}\frac{Q_\beta-1}{2}\right)}\frac{P}{Q}.$$

Si ahora hacemos que

$$\frac{P-1}{2}=\sum_{\alpha=1}^{r}\frac{P_\alpha-1}{2}+2N \text{ y } \frac{Q-1}{2}=\sum_{\beta=1}^{s}\frac{Q_\beta-1}{2}+2N_1,$$

entonces:

$$\left(\frac{Q}{P}\right)=(-1)^{(P-1)(Q-1)/4}\left(\frac{P}{Q}\right).$$

Capítulo 2

ElGammal y el logaritmo discreto

2.1. Generadores

2.131 Definition Let $\alpha \in \mathbb{Z}_n^*$. If the order of α is $\phi(n)$, then α is said to be a *generator* or a *primitive element* of $\mathbb{Z}_n^*$. If $\mathbb{Z}_n^*$ has a generator, then $\mathbb{Z}_n^*$ is said to be *cyclic*.

Figura 2.1: [1]

2.132 Fact (*properties of generators of* $\mathbb{Z}_n^*$)

(i) $\mathbb{Z}_n^*$ has a generator if and only if $n = 2, 4, p^k$ or $2p^k$, where p is an odd prime and $k \geq 1$. In particular, if p is a prime, then $\mathbb{Z}_p^*$ has a generator.

(ii) If α is a generator of $\mathbb{Z}_n^*$, then $\mathbb{Z}_n^* = \{\alpha^i \bmod n \mid 0 \leq i \leq \phi(n) - 1\}$.

(iii) Suppose that α is a generator of $\mathbb{Z}_n^*$. Then $b = \alpha^i \bmod n$ is also a generator of $\mathbb{Z}_n^*$ if and only if $\gcd(i, \phi(n)) = 1$. It follows that if $\mathbb{Z}_n^*$ is cyclic, then the number of generators is $\phi(\phi(n))$.

(iv) $\alpha \in \mathbb{Z}_n^*$ is a generator of $\mathbb{Z}_n^*$ if and only if $\alpha^{\phi(n)/p} \not\equiv 1 \pmod{n}$ for each prime divisor p of $\phi(n)$.

Figura 2.2: [1]

Primero que nada hay que probar que 2 es un generador en $\mathbb{Z}_{2027}^*$

Para hacerlo (y por definición) podemos recordar que un generador es si el orden de 2 sea igual a $\phi(n)$.

Esto nos lleva a la definición de orden, que esta definido como la t mas pequeña tal que $2^t \equiv 1 \mod 2027$.

Esto lo podemos probar con fuerza bruta, y un pequeño programa en Python:

```
alpha = 2
t = 1
mod = 2027

while pow(alpha, t, mod) != 1:
  t += 1

print (t)
```

Llegando a que $t = 2026$ y dado que 2027 es un primo es obvio que $\phi(p) = p - 1 = 2027 - 1 = 2026$. Por lo tanto la demostración por fuerza bruta esta lista.

2.1.1. Otra forma

Otra forma de llegar a esto es usando la idea del Menezes de que: $\alpha \in \mathbb{Z}^* n$ es un generador si y solo si $\alpha^{\frac{\phi(n)}{p}} \not\equiv 1 \mod n$ para cada factor primero de $\phi(n)$.

Ahora como dijimos $\phi(p) = p - 1 = 2027 - 1 = 2026 = 2 * 1013$, por lo tanto vamos a ir por sus factores primos:

- 2

  Ahora vamos:

  $$\begin{aligned} \alpha^{\frac{\phi(n)}{p}} &\equiv \alpha^{\frac{2026}{2}} \pmod{2027} \\ &\equiv \alpha^{1013} \pmod{2027} \\ &\equiv 2^{1013} \pmod{2027} \\ &\equiv 2026 \pmod{2027} \end{aligned}$$

  Y mira no es uno, por lo que vamos bien.

- 1013

Ahora vamos:

$$\begin{aligned}\alpha^{\frac{\phi(n)}{p}} &\equiv \alpha^{\frac{2026}{1013}} \pmod{2027} \\ &\equiv \alpha^2 \pmod{2027} \\ &\equiv 2^2 \pmod{2027} \\ &\equiv 4 \pmod{2027}\end{aligned}$$

Y mira no es uno, por lo que todo salio bien.

Por lo tanto 2 es un generador. QED

2.2. Calculo de indices

2.1 Select a random integer k, $0 \leq k \leq n-1$, and compute α^k.

2.2 Try to write α^k as a product of elements in S:

$$\alpha^k = \prod_{i=1}^{t} p_i^{c_i}, \quad c_i \geq 0. \tag{3.5}$$

If successful, take logarithms of both sides of equation (3.5) to obtain a linear relation

$$k \equiv \sum_{i=1}^{t} c_i \log_\alpha p_i \pmod{n}. \tag{3.6}$$

2.3 Repeat steps 2.1 and 2.2 until $t + c$ relations of the form (3.6) are obtained (c is a small positive integer, e.g. $c = 10$, such that the system of equations given by the $t + c$ relations has a unique solution with high probability).

3. (*Find the logarithms of elements in S*) Working modulo n, solve the linear system of $t + c$ equations (in t unknowns) of the form (3.6) collected in step 2 to obtain the values of $\log_\alpha p_i$, $1 \leq i \leq t$.

4. (*Compute y*)

4.1 Select a random integer k, $0 \leq k \leq n-1$, and compute $\beta \cdot \alpha^k$.

4.2 Try to write $\beta \cdot \alpha^k$ as a product of elements in S:

$$\beta \cdot \alpha^k = \prod_{i=1}^{t} p_i^{d_i}, \quad d_i \geq 0. \tag{3.7}$$

If the attempt is unsuccessful then repeat step 4.1. Otherwise, taking logarithms of both sides of equation (3.7) yields $\log_\alpha \beta = (\sum_{i=1}^{t} d_i \log_\alpha p_i - k) \bmod n$; thus, compute $y = (\sum_{i=1}^{t} d_i \log_\alpha p_i - k) \bmod n$ and return(y).

Figura 2.3: [1]

Ok, lo bueno es que ya me dieron la base: $2, 3, 5, 7, 11$.

Ahora vamos a obtener 6 relaciones que fueron exitosas (dejo fuera las que fueron mal):

- $2^{769} \equiv 539 \equiv 11 * 7^2 \mod 2027$
- $2^{1322} \equiv 231 \equiv 11 * 7 * 3 \mod 2027$

- $2^{1912} \equiv 25 \equiv 5 * 5 \mod 2027$
- $2^{1756} \equiv 14 \equiv 7 * 2 \mod 2027$
- $2^{857} \equiv 567 \equiv 7 * 3^4 \mod 2027$
- $2^{1799} \equiv 1250 \equiv 5^4 * 2 \mod 2027$

Lo bueno es que hacer $2^x \mod 2027$ es $O(log_2(n))$

Con estas relaciones podemos escribir las siguiente ecuaciones:

- $769 \equiv log_2(11) + 2log_2(7) \mod 2026$
- $1322 \equiv log_2(7) + 2log_2(3) \mod 2026$
- $1912 \equiv 2log_2(5) \mod 2026$
- $1756 \equiv 2log_2(7) \mod 2026$
- $857 \equiv log_2(7) + 4log_2(3) \mod 2026$
- $1799 \equiv 4log_2(5) + log_2(2) \mod 2026$

Ahora, podemos ver esto como:

- $769 \equiv e + 2d \mod 2026$
- $1322 \equiv d + 2b \mod 2026$
- $1912 \equiv 2c \mod 2026$
- $1756 \equiv 2d \mod 2026$
- $857 \equiv d + 4b \mod 2026$
- $1799 \equiv 4c + 1 \mod 2026$

Ahora tras un poco de algebra bastante trivial podemos llegar a esto:

- $2, 3, 5, 7, 11$
- $log_2(2) \equiv 1 \mod 2026$

- $log_2(3) \equiv 282 \mod 2026$
- $log_2(5) \equiv 1969 \mod 2026$
- $log_2(7) \equiv 1755 \mod 2026$
- $log_2(11) \equiv 1311 \mod 2026$

Ahora vamos a seleccionar enteros a lo random, despues de intentarlo un rato llegue a esto:

$13*2^{1340} \equiv 550 \equiv 13*666 \equiv 550 \mod 2027$ y tenemos que $5^2*2*11 \equiv 550$ mód 2027

Ahora si que podemos escribir que:

$$\begin{aligned} log_2(13) &\equiv (2log_2(5) + log_2(2) + log_2(11) - 1340) \mod 2026 \\ &\equiv (2(1969) + 1 + 1311 - 1340) \mod 2026 \\ &\equiv 1884 \quad (\text{mód } 2026) \end{aligned}$$

Y podemos comprobar rápido que $2^{1884} \equiv 13 \mod 2027$

2.3. ElGammal

SUMMARY: B encrypts a message m for A, which A decrypts.

1. *Encryption.* B should do the following:
   (a) Obtain A's authentic public key (p, α, α^a).
   (b) Represent the message as an integer m in the range $\{0, 1, \ldots, p-1\}$.
   (c) Select a random integer k, $1 \leq k \leq p-2$.
   (d) Compute $\gamma = \alpha^k \bmod p$ and $\delta = m \cdot (\alpha^a)^k \bmod p$.
   (e) Send the ciphertext $c = (\gamma, \delta)$ to A.
2. *Decryption.* To recover plaintext m from c, A should do the following:
   (a) Use the private key a to compute $\gamma^{p-1-a} \bmod p$ (note: $\gamma^{p-1-a} = \gamma^{-a} = \alpha^{-ak}$).
   (b) Recover m by computing $(\gamma^{-a}) \cdot \delta \bmod p$.

Figura 2.4: [1]

Ahora que tenemos la llave privada descifrar el mensaje es bastante sencillo, pero aun asi para que se entienda vamos por pasos:

- Por un lado lo que tenemos es una serie de pares por lo que podemos imaginar que cifraron caracter por caracter, asi que vamos a solucionar un caracter y el resto será analogo.

- Ahora lo bueno es que justo el dia que escribo esto nos toco hacer ElGammal en el laboratio por la idea esta muy fresca.

  Tu tienes un primo p (2027), eliges un generador (2), y un $2^k = 13$, estas son llaves publicas y justo la k es la privada (ya vimos que es 1884), ahora, dado un mensaje (y, δ) para descrifrar lo que hacemos es $y^{p-1-a} * \delta \mod p$ esto sale rapido:

- Por ejemplo para el primero mensaje tenemos que:

$$\begin{aligned} y^{p-1-a} * \delta \mod p &\equiv 128^{2027-1-1884} * 793 \mod 2027 \\ &\equiv 4 \pmod{2027} \end{aligned}$$

  por lo tanto la primera letra del mensaje es una 'e'.

- Ahora vamos a simular el proceso que para algo somos computologos:

```
mod = p = 2027
a = 1884

cipher = [
    (128, 793),   (128, 528),  (128, 1233),  (128, 793),
    (128, 793),  (128, 264),  (128, 793),  (128, 1850),
    (128, 1410),  (128, 1586),  (128, 1410),  (128, 1586),
    (128, 1762),  (128, 793),  (128, 528),
    (128, 1233),  (128, 87),  (128, 352),  (128, 1938),
    (128, 704),  (128, 1498),  (128, 87),
    (128, 1410),  (128, 1586),  (128, 1674),
    (128, 87),  (128, 1674),  (128, 1586),  (128, 176),
    (128, 1938),  (128, 87),  (128, 1674),  (128, 1145),
    (128, 1938),  (128, 793),  (128, 1674),
    (128, 87),    (128, 1233),  (128, 87),  (128, 1850),
    (128, 793),  (128, 87),
]


def solve(y: int, delta:int) -> int:
    m = (pow(y, p - 1 - a, mod) * delta) % mod
    m = m % 26
    m += ord("a")  # works with ASCII

    return chr(m)


message = "".join([solve(y, delta) for y, delta in cipher])
print(message)
```

Dando el mensaje:

```
esteejercicioestamuyfacilaligualquelatarea
```

Que buen mensaje.

Capítulo 3

Criba cuadratica y RSA

3.1. Criba cuadratica

Primero que nada hay que seleccionar una base de factores, en este caso solo vamos a poner a los primos p para los cuales n es un residuo cuadratico modulo p: $S = Base = B = \{ -1, 2, 3, 13, 17, 19, 29 \}$.

Esto se hace con un sencillo programita:

```
from sympy.ntheory import legendre_symbol

def is_prime(n):
    if (n <= 1):
        return False
    if (n <= 3):
        return True

    if (n % 2 == 0 or n % 3 == 0):
        return False

    i = 5
    while (i * i <= n):
        if (n % i == 0 or n % (i + 2) == 0):
            return False
        i = i + 6

    return True


n = 87463
for i in range(3, 50):
    if is_prime(i) and legendre_symbol(n, i) == 1:
        print(i)
```

Ahora lo siguiente es calcular $M = m = \lfloor \sqrt{n} \rfloor$, que en este caso 295, finalmente vamos a calcular la tabla de la famosa criba cuadratica:

i	x	$b = q(x) = (x+m)^2 - n$	factors	a_i	v_i
1	1	$(296)^2 - 87463 = 153$	$3^2 * 17$	296	(0, 0, 0, 0, 1, 0, 0)
2	4	$(299)^2 - 87463 = 1938$	$2 * 3 * 17 * 19$	299	(0, 1, 1, 0, 1, 1, 0)
3	12	$(307)^2 - 87463 = 6786$	$2 * 3^2 * 13 * 29$	307	(0, 1, 0, 1, 0, 0, 1)
4	-17	$(278)^2 - 87463 = -10179$	$-1 * 3^3 + 13 + 29$	278	(1, 0, 1, 1, 0, 0, 1)
4	21	$(316)^2 - 87463 = 12393$	$3^6 * 17$	316	(0, 0, 0, 0, 1, 0, 0)
5	-30	$(265)^2 - 87463 = -17238$	$-1 * 2 * 3 * 13 * 17$	265	(1, 1, 1, 1, 1, 0, 0)

Ahora, podriamos seguir haciendolo pero mira que ya tenemos que $a_1 + a_4 = 0$, asi que vamos a probarla:

Ademas creo importante añadir que no hice estas cuentas solo a mano, llegue a hacer hasta 12 a mano, pero despues me di por vencido y mejor programe lo que estaba haciendo una y otra vez:

```
from math import floor, sqrt
from sympy import primefactors, factorint

primes = [-1, 2, 3, 13, 17, 19, 29]
n = 87463
m = floor(sqrt(n))

i = 1
x = 1
while (i <= 7):
    ai = x + m
    b = ai * ai - n

    factors = factorint(b, limit=29)
    if all([f <= primes[-1] for f in factors.keys()]) and
   len(factors) > 1:

        print(f"i={i} \t ai={ai} \t x={x} \t", end="")
        print(f"{b} = ", end="")
        for prime, exponent in factors.items():
            print(f"{prime}^{exponent} * ", end="")
        print()

        i += 1

    x = -x + 1 if x < 0 else -x
```

Obteniendo esta tabla:

```
i=1       ai=296    x=1     153      = 3^2 * 17^1
i=2       ai=299    x=4     1938     = 2^1 * 3^1 * 17^1 * 19^1
i=3       ai=307    x=12    6786     = 2^1 * 3^2 * 13^1 * 29^1
i=4       ai=278    x=-17   -10179   = 3^3 * 13^1 * 29^1 * -1^1
i=5       ai=316    x=21    12393    = 3^6 * 17^1
i=6       ai=265    x=-30   -17238   = 2^1 * 3^1 * 13^2 * 17^1 * -1^1
i=7       ai=347    x=52    32946    = 2^1 * 3^1 * 17^2 * 19^1
```

Ahora, si, regresemos a lo que estabamos viendo, que $x = a_1 * a_4 = 296 * 316 = 6073 \mod 87463$

Ahora hagamos las $l's$:

- $l_1 = 0$
- $l_2 = 0$
- $l_3 = 4$
- $l_4 = 0$
- $l_5 = 1$
- $l_6 = 0$
- $l_7 = 0$

Con estos damos podemos ya calcular a: $y = 3^4 * 17 = 1377 \mod 87463$.

Ahora basta con sacar el $587 = gcd(6073 - 1377, 87463)$ por lo que tenemos que 587 es un factor no trivial, de hecho ya con este es obvio que:

$87463 = 587 * 149$

3.2. Ahora vamos con RSA

Ahora veamos que los parametros publicos son $(87463, 15157)$, es decir (n, e) y lo que estamos buscando es la llave privada (d) tal que $ed \equiv$ mód $\phi(n)$ ahora sacar el inverso es sencillo si sabes la factorización de n, porque $n = p * q$, entonces $\phi(n) = (p-1) * (q-1) = 586 * 148 = 86728$, por lo que un simple algoritmo extendido de euclides nos da el inverso, dando que $d = 50485$.

Con esta llave podemos descifrar todo muy sencillo:

Es mas hice un programa que usando las ideas bases del algoritmo te ayuda con la pesada tarea de descifrarlo uno a uno, despues de todo justo este fue un proyecto de laboratorio.

```
from Crypto.Util.number import getPrime, GCD, inverse
from math import log2

big_length = int(log2(1e100))


class RSA:
    def __init__(self):
        p = getPrime(big_length)
        q = getPrime(big_length)

        self.n = p * q

        phi_n = (p - 1) * (q - 1)
        public_key = phi_n - 2

        while GCD(phi_n, public_key) != 1:
            public_key -= 1

        self.private_key = inverse(public_key, phi_n)
        self.public_key = public_key

    def get_keys(self):
        return self.n, public_key

    def encrypt(self, plaintext):
        return [pow(ord(char), self.public_key, self.n) for char in plaintext]

    def dencrypt(self, cipher):
        plain = [pow(char, self.private_key, self.n) % 26 for char in cipher]
        plain = [chr(m + ord("a")) for m in plain]

        return "".join(plain)


solver = RSA()
solver.n = 87463
solver.private_key = 50485
```

```
solver.public_key = 15157

cipher = [
    21347, 41185, 31564, 41185, 76237, 73700, 53597, 21347, 31564, 73700,
    21347, 73700, 53597, 14144, 42561, 73700, 53597, 73593, 14420, 76237,
    41185, 76237, 23637, 14420, 1, 31564, 41185, 14420, 76237, 2136,
    41185, 22481, 21347, 73700, 73593, 14420, 76237, 73700, 53597, 82282,
    19930, 22481, 14420, 31564, 73700, 53597, 31564, 14420, 41185, 76237,
    14420, 53597, 82282, 73700, 14420, 53597, 53597, 19930, 67024, 14144,
    2136, 14144, 14420, 82282, 42561, 14420,
]

message = solver.dencrypt(cipher)

print(message)
```

Y llegamos a esto:

```
paralospropositosdelalgebraelcampodelosnumerosrealesnoessuficiente
```

Capítulo 4

Generadores

Vamos a mostrar que el problema del logaritmo discreto no depende del generador.

Sea α and γ dos generadores de un grupo cíclico G de orden n, y sea $\beta \in G$. Sea $x = log_\alpha \beta$, $y = log_\gamma \beta$, y $z = log_\alpha \gamma$.

Entonces $\alpha^x = \beta = \gamma^y = (\alpha^z)^y$. Por consiguiente $x = zy \mod n$, y

$$log_\gamma \beta = (log_\alpha \beta)(log_\alpha \gamma)^{-1} \mod n$$

Esto significa que cualquier algoritmo de que calcule logarítmos para una base α puede ser usado para calcular logaritmos a cualquier otra base γ que también es un generador de G.

Bibliografía

[1] HANDBOOK of APPLIED CRYPTOGRAPHY *Alfred J. Menezes* .