Proyecto CLCript
Cuadernos de Laboratorio de Criptografía. Entrega nº 6. Última actualización 06/05/19
Autor: Dr. Jorge Ramió Aguirre (@criptored)
Prácticas con el algoritmo RSA: números no cifrables NNC

- Software genRSA v2.1: https://www.criptored.es/software/sw_m001d.htm
- Lectura de interés: https://www.criptored.es/crypt4you/temas/RSA/leccion5/leccion05.html
- Lectura de interés: https://en.wikipedia.org/wiki/Safe_prime

Objetivos:

1. Observar la cantidad de NNC de una clave RSA en función de los primos p y q.
2. Comprobar que no es una tarea sencilla encontrar los NNC si el módulo es grande.
3. Comprobar que los NNC tienen una distribución característica dentro del cuerpo de cifra n.
4. Comprobar que con primos seguros se minimiza la cantidad de NNC.
5. Comprobar que la cantidad máxima de NNC puede llegar a ser el cuerpo de cifra completo.
6. Comprobar que los NNC no se traducen en una vulnerabilidad en RSA.
7. Observar curiosidades y particularidades de los NNC en RSA.

I. Cantidad de NNC en RSA

Ejercicio 1) NNC con primos no seguros

1.1. Con genRSA v2.1 genera de forma Manual estas 4 claves de 32, 64, 128 y 1.024 bits en formato decimal. Observa que todas tienen más de mil números no cifrables.
p = 41.761, q = 51.521, e = 65.537
p = 2.431.891.457, q = 3.969.189.899, e = 65.537
p = 12.519.884.402.137.594.369, q = 17.669.745.194.824.060.361, e = 65.537
p =
12.053.134.069.622.801.747.370.057.646.463.199.590.073.795.578.680.365.701.862.00
8.743.842.493.806.489.196.671.032.909.147.154.537.666.904.373.227.731.769.008.042
.979.204.805.969.521.119.763.440.008.328.097,
q =
8.784.294.713.643.489.507.733.312.398.741.999.681.651.513.735.558.027.327.026.612
.992.999.379.038.861.684.153.247.644.211.017.747.201.206.045.620.919.427.701.122.
565.881.266.592.118.942.947.001.004.938.041
e = 65.537

1.2. Si lo que se cifra con la clave pública e del destinatario es un número K secreto, por ejemplo una clave de sesión para un algoritmo de cifra simétrica, ¿qué sucedería si el valor K elegido fuese precisamente un número no cifrable?

1.3. Y si el módulo n del destinatario es grande, e.g. 2.048 bits, y la clave de sesión K es un número aleatorio de 256 bits, ¿se detectaría este hecho en el criptograma recibido?

Comprueba tu trabajo:

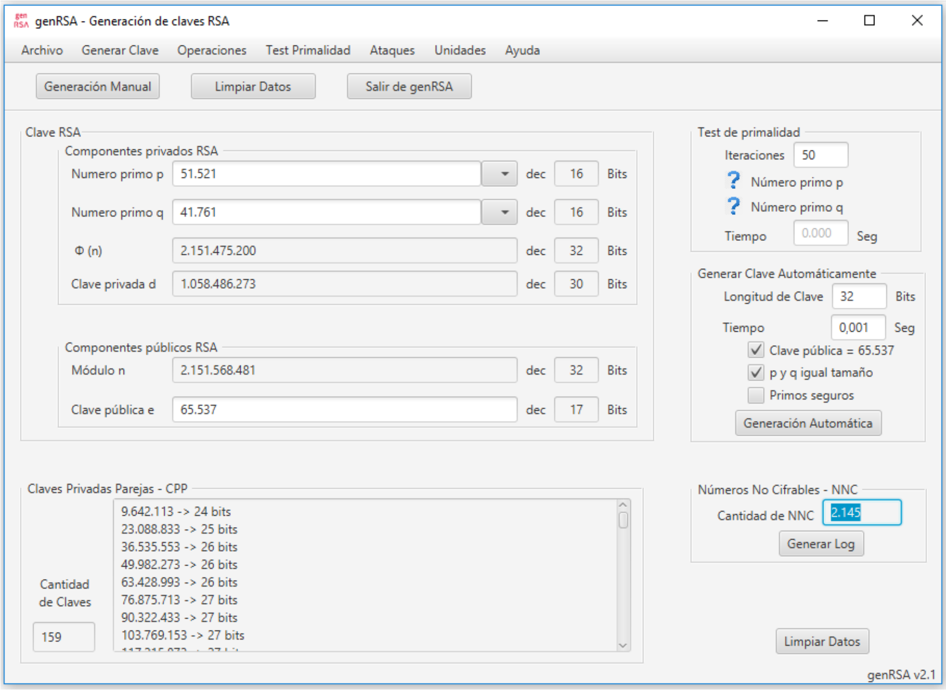

Figura 1. Clave de 32 bits con más de 1.000 NNC.

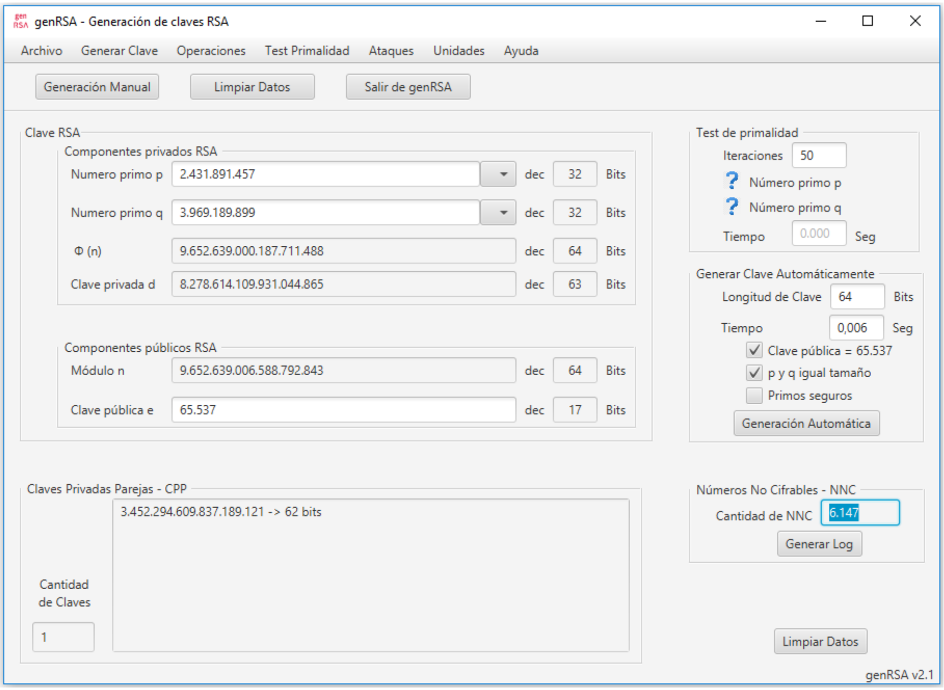

Figura 2. Clave de 64 bits con más de 1.000 NNC.

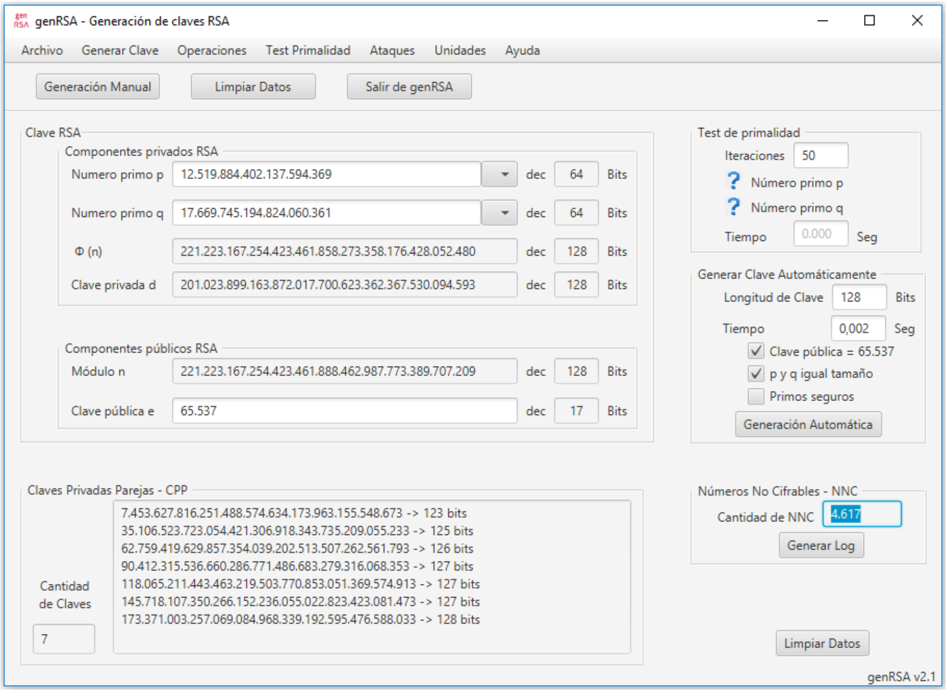

Figura 3. Clave de 128 bits con más de 1.000 NNC.

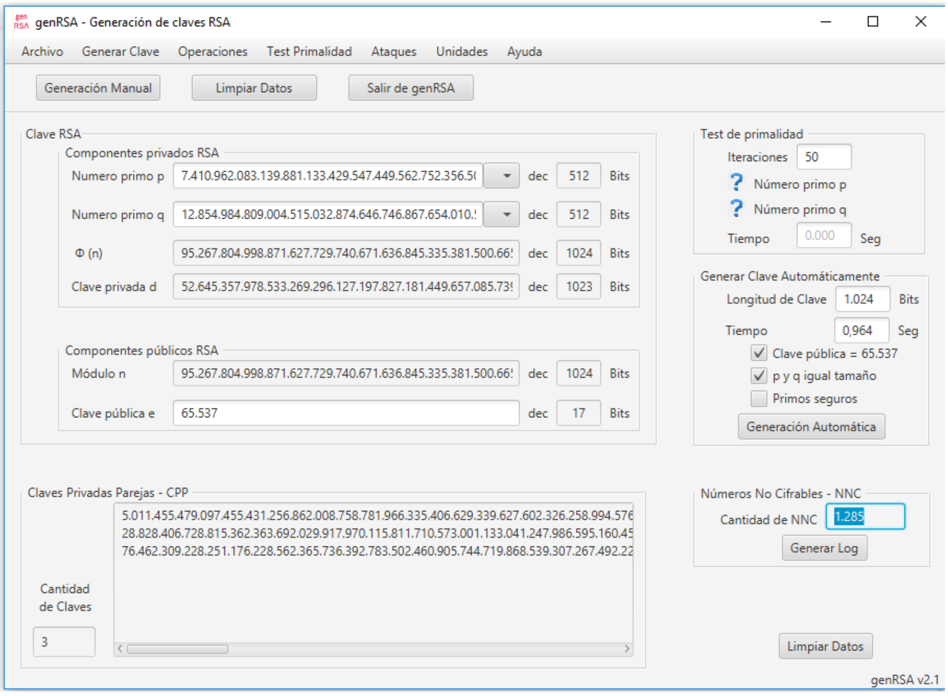

Figura 4. Clave de 1.024 bits con más de 1.000 NNC.

Ejercicio 2) NNC con primos seguros

2.1. Con genRSA v2.1 genera de forma Manual estas 4 claves de 32, 64, 128 y 1.024 bits en formato decimal. Observa que todas tienen 9 números no cifrables.

p = 35.963, q = 65.123, e = 65.537

p = 4.086.747.383, q = 3.844.366.139, e = 65.537

p = 14.709.964.030.266.707.243, q = 16.962.152.053.789.837.463, e = 65.537

p = 9.314.731.805.585.770.091.834.655.572.182.860.162.350.235.845.548.452.191.147.471

.272.512.110.960.778.774.936.976.696.175.968.589.945.530.298.063.252.425.716.967.
893.545.700.521.782.589.802.578.943.586.247
q =
11.170.398.797.211.189.170.652.191.375.164.287.178.556.626.496.466.672.635.653.19
7.249.050.068.544.314.436.777.034.494.931.015.478.332.036.853.064.961.310.909.242
.258.088.817.917.550.457.280.909.856.429.867
e = 65.537

2.2. Comprueba que generar de forma Automática una clave de 1.024 bits con primos seguros es una operación que puede tardar varios segundos o minutos, y que si no se usan primos seguros dicha operación es muy rápida.

2.3. ¿Por qué generar una clave de forma Automática de 2.048 bits con primos seguros tarda tanto tiempo y, en cambio, si la clave se genera de forma Manual, el resultado es inmediato?

Comprueba tu trabajo:

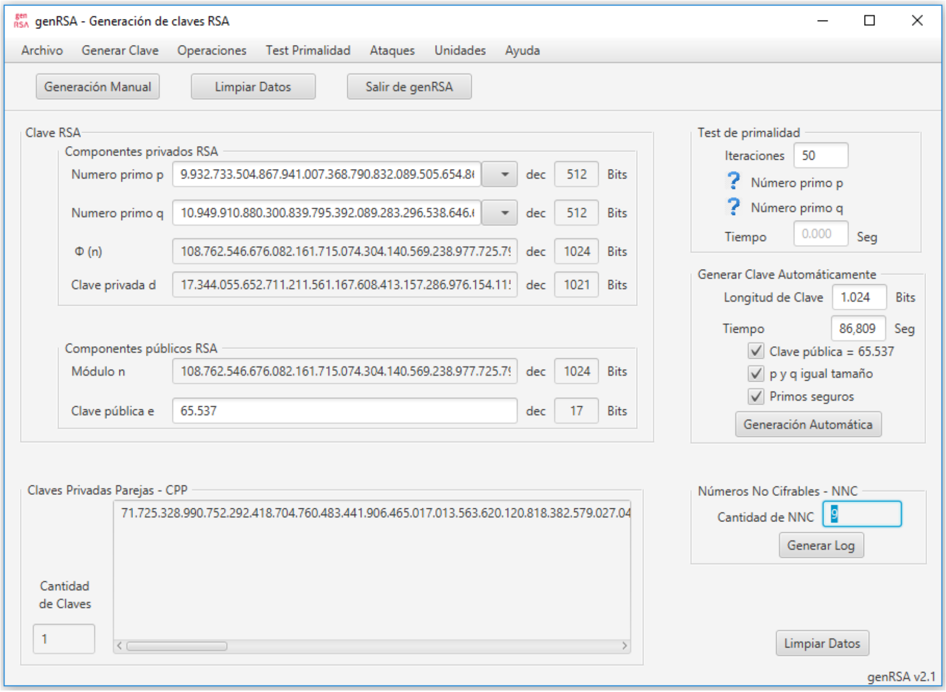

Figura 5. Clave de 1.024 bits del apartado 2.1.5.

II. Cálculo de los NNC en RSA

Ejercicio 3) Cálculo de los NNC

3.1 Encuentra los 121 NNC de la clave RSA de 32 bits con p = 52.571, q = 61.651 y e = 11, creando la clave de forma Manual y luego Generando el Log correspondiente. Como esta operación va a requerir ataques a los valores de p y q, necesitaremos un cierto tiempo de cómputo. No obstante, como aquí p y q son muy pequeños, de 16 bits cada uno, ese Log deberá encontrarse en un par de segundos.

3.2 Abre el archivo Log de NNC con un navegador y comprueba que están los números 0, 1 y n-1.

3.3 Comprueba que aparecen números de 8, 9 y 10 dígitos y que la distribución de esos números se asemeja a una distribución uniforme continua.

3.4 Repite el ejercicio para una clave de 50 bits: p = 27.966.577, q = 32.577.319 y e = 65.537. Observa que ahora el log ha tardado casi 5 minutos para encontrar los 51 NNC de la clave porque p y q tienen ahora 25 bits en vez de 16 bits.

3.5 ¿Se sigue manteniendo esa distribución de números uniforme?

3.6 Con el tiempo medido en el apartado 2.4, ¿a qué tasa media aproximada de cifrados por segundo ha trabajo el programa? Con ese dato, ¿qué tiempo tardaría en encontrar los NNC de una clave de 128 bits? ¿Será posible encontrar los NNC de una clave real de 2.048 bits?

Comprueba tu trabajo:

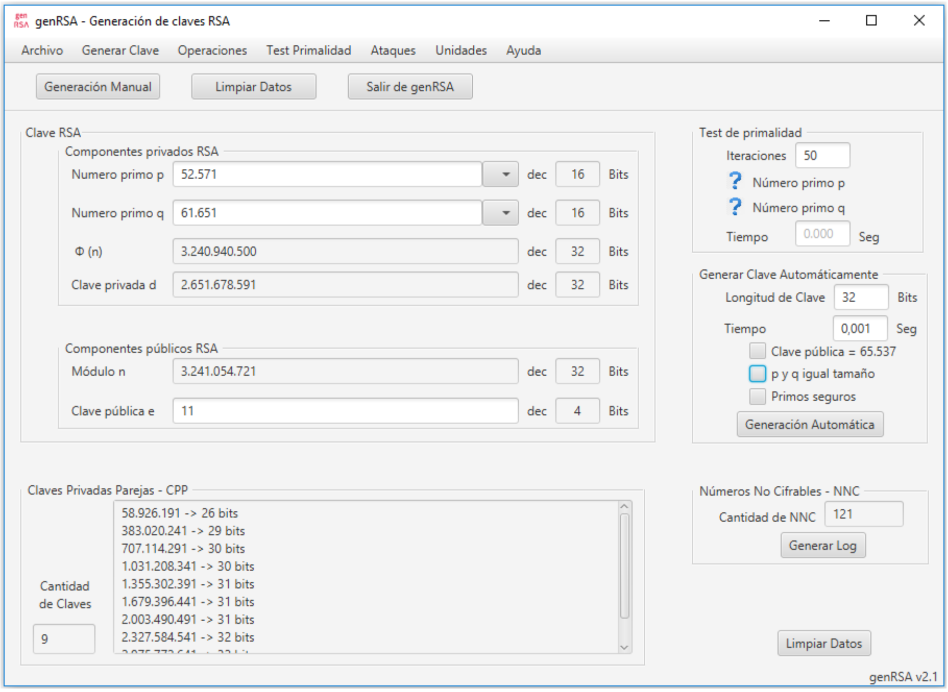

Figura 6. Clave de 32 bits del apartado 3.1.

Documento LOG de NNC genRSA v2.1
Número primo P generado: 52.571
Número primo Q generado: 61.651
Módulo N generado: 3.241.054.721
Clave Pública e generada: 11
Clave Privada d generada: 2.651.678.591
NÚMEROS NO CIFRABLES
La cantidad de Números No Cifrables es: 121

0	...
1	3.089.023.355
12.206.899	3.095.065.055
13.563.318	3.101.000.076
19.605.017	3.102.586.673
68.179.086	3.107.271.953
106.656.132	3.108.628.372
118.863.030	3.108.628.373
120.219.450	3.120.835.271
132.426.348	3.122.191.691
132.426.349	3.134.398.589
133.782.768	3.172.875.635
138.468.048	3.221.449.704
140.054.645	3.227.491.403
145.989.666	3.228.847.822
...	3.241.054.720

Figura 7. Log de NNC de la clave de 32 bits del apartado 3.1.

Ejercicio 4) Minimizando y maximizando los NNC

4.1. Al usar primos seguros, los NNC tienden a minimizarse en el valor 9 pero esto depende también de la clave pública e. Comprueba que si generas de forma Automática claves de diferentes tamaños (e.g. de 32 bits) con primos seguros y el valor menor posible de e, o bien el número 4 de Fermat, siempre se obtienen 9 NNC, el mínimo.

4.2. Genera de forma Manual la clave decimal de 24 bits con p = 3.167, q = 3.779, e = 3. Cambia el valor de e = 5, 7, 9, 11 y observa que siempre se obtienen 9 NNC. Usa ahora los valores de e = 3.167 (el primo p), e = 3.779 (el primo q) y observa la cantidad de NNC.

4.3. Usa ahora e = 5.980.575 y observa que todo el cuerpo es no cifrable.

4.4. Genera una clave de 20 bits con p = 673, q = 967 (primos no seguros, compruébalo), e = 5 y observa sus NNC. Genera ahora la clave tomando como valor e = [ϕ(n)/k + 1], con k = 2, 3, 4, 5, 6, 7, 8, 9, 10, que sean válidos. Observa cómo varía la cantidad de NNC.

k = 2	e = [ϕ(n)/2 + 1] = (649.152/2 + 1)	e = 324.577 (clave válida)
k = 3	e = [ϕ(n)/3 + 1] = (649.152/3 + 1)	e = 216.385 (clave válida)
k = 4	e = [ϕ(n)/4 + 1] = (649.152/4 + 1)	e = 162.289 (clave válida)
k = 5	e = [ϕ(n)/5 + 1] = (649.152/5 + 1)	Número con decimales
k = 6	e = [ϕ(n)/6 + 1] = (649.152/6 + 1)	e = 108.193 (clave válida)
k = 7	e = [ϕ(n)/7 + 1] = (649.152/7 + 1)	e = 92.737 (clave válida)
k = 8	e = [ϕ(n)/8 + 1] = (649.152/8 + 1)	e = 81.145 (clave válida)
k = 9	e = [ϕ(n)/9 + 1] = (649.152/9 + 1)	e = 72.129 (clave NO válida)
k = 10	e = [ϕ(n)/10 + 1] = (649.152/10 + 1)	Número con decimales

4.5. Comprueba que, cuando todo el cuerpo es no cifrable, algunas veces la clave pública tiene el mismo valor que la clave privada y que, además, aparece una clave privada extra con el valor 1.

Comprueba tu trabajo:

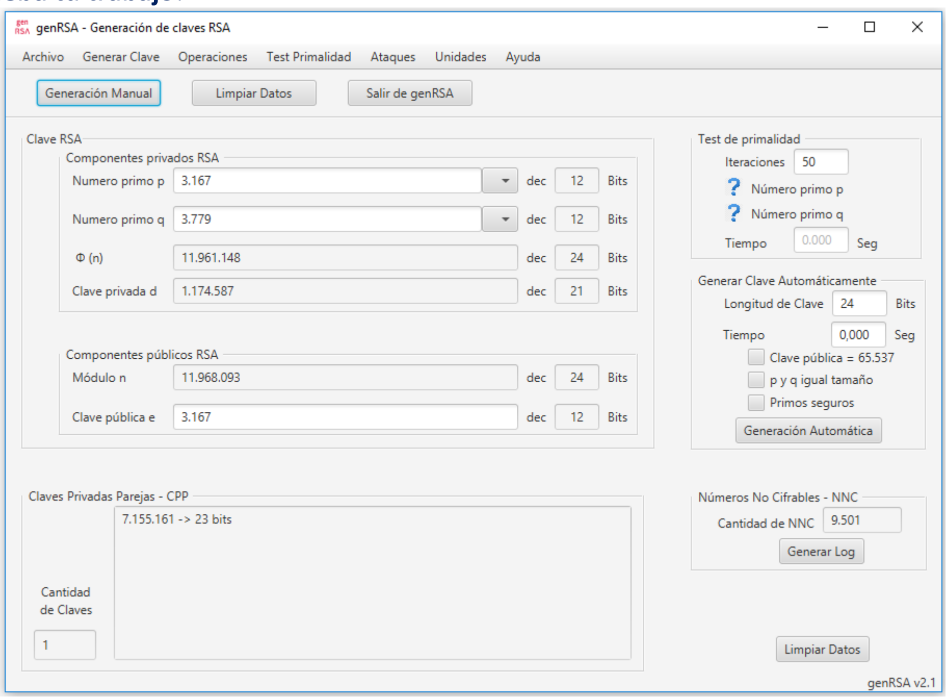

Figura 8. Clave RSA de 24 bits.

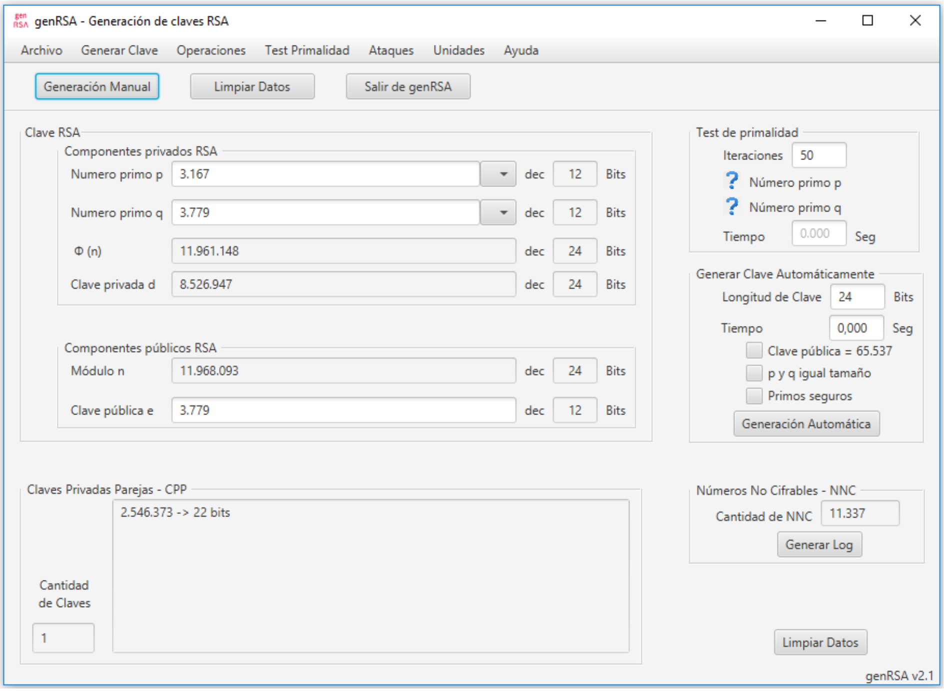

Figura 9. Clave de 24 bits en que la clave e toma los valores de los primos p o q.

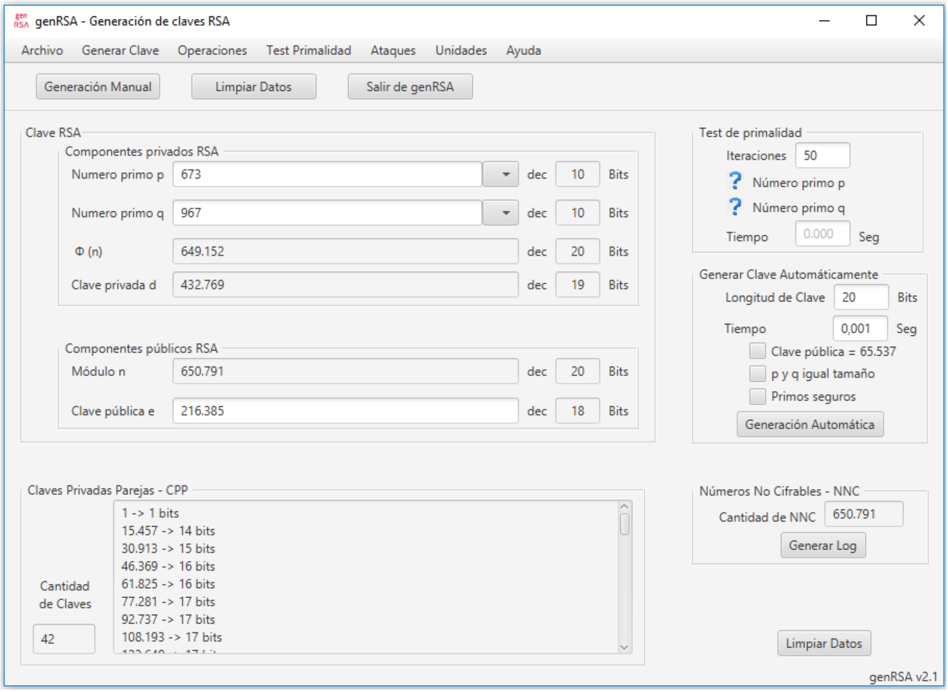

Figura 10. Clave de 20 bits con e = [ϕ(n)/3 + 1] = (649.152/3 + 1) = 216.385: NNC = n.

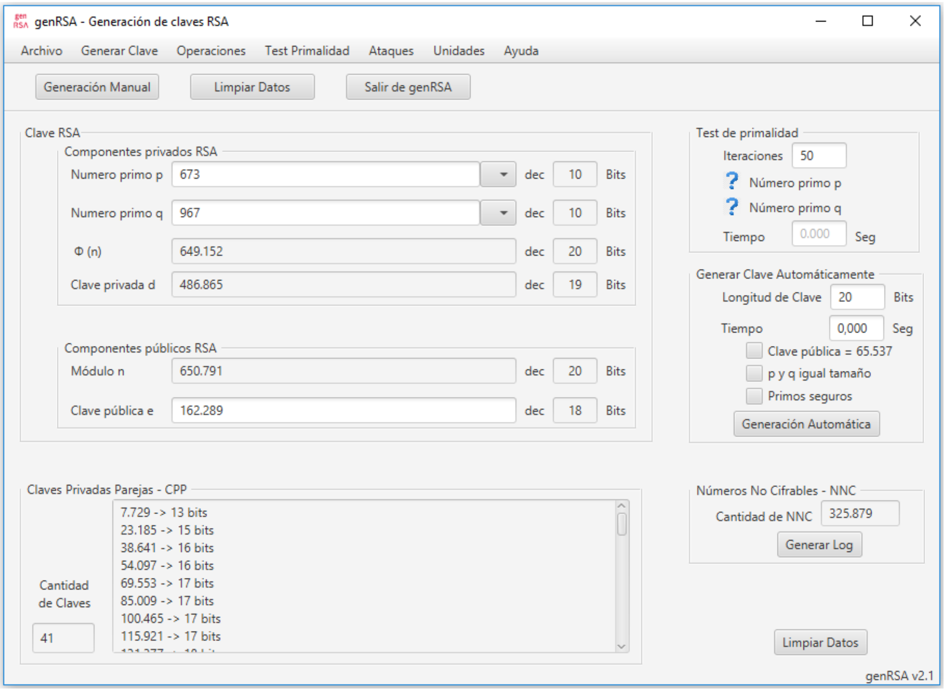

Figura 11. Clave de 20 bits con e = [ϕ(n)/4 + 1] = (649.152/4 + 1) = 162.289.

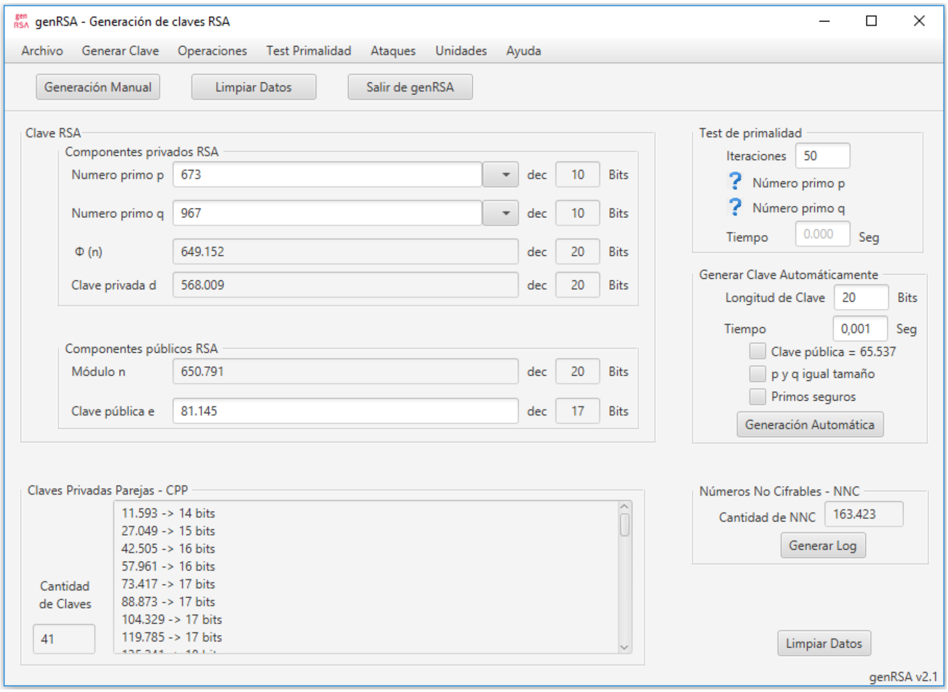

Figura 12. Clave de 20 bits con e = [ϕ(n)/8 + 1] = (649.152/8 + 1) = 81.145.

Ejercicio 5) Particularidades de los NNC

5.1. Genera de forma Manual una clave decimal de 16 bits con p = 193, q = 241, e = 7. Genera el Log de los NNC, observa que hay 49 NNC y que varios de ellos están separados en una unidad, como es el caso de:
0 – 1; 964 – 965; 3.389 - 3.390; 11.085 - 11.086; 12.050 - 12.051; 14.475 - 14.476; 19.987 - 19.988; 26.525 - 26.526; 31.073 - 31.074; 32.037 - 32.038; 34.462 - 34.463; 35.427 - 35.428; 43.123 - 43.124; 45.548 - 45.549

5.2. Observa ahora que, excepto el valor 0 inicial, los extremos de esa cadena de 49-1 = 48 números no cifrables siempre suman el módulo n = 46.513:
1 + 46.512 = 964 + 45.549 = 965 + 45.548 = 1.929 + 44.584 = ... 20.952 + 25.561 = 46.513

5.3. ¿Crees que estas dos singularidades pueden ser una debilidad de RSA, en el sentido de que pudiera conocerse de antemano algún número no cifrable y forzar así a que se use este número como valor secreto a cifrar?

5.4. Genera de forma Automática varias claves de diferentes tamaños, por ejemplo 10, 20, 30, 40 y 50 bits, activando solamente la opción p y q de igual tamaño. Como en este caso la clave pública e que elige el programa será siempre el menor número que cumpla la condición, comprueba que la cantidad de NNC son siempre unos pocos números que, además, se repiten: 9, 15, 21, 25, 33, 35, 39, 49, 51, 91, 121, 123, etc.

5.5. ¿Qué te sugiere esto?

5.6. Encuentra los Números No Cifrables de las siguientes claves con:
Clave a) p = 1.013, q = 1.021, e = 7
Clave b) p = 521, q = 1.009, e = 11
Clave c) p = 691, q = 877, e = 7
Clave d) p = 1.013, q = 1.021, e = 13

5.7. Observa que, dejando de lado el 0, la mitad de esos números son pares y la otra mitad son impares.

5.8. ¿Tiene esto algo que ver con que la suma de los extremos (excepto el 0) de la lista dé siempre como resultado el valor de n, y el módulo n deba ser obligatoriamente un número impar por ser el producto de dos primos?

5.9. Comprueba que, a medida que aumenta el tamaño del cuerpo n, los NNC van perdiendo esa apariencia de distribución uniforme dentro del cuerpo y que, excepto el 0 y el 1, los demás NNC se ubican muy cerca del módulo n. Para ello, genera de forma automática claves RSA de 10, 20, 30, 40 y 50, con clave pública e pequeña (no F4) y que cada clave tenga 33 NNC. Con pocos intentos lo lograrás, forzando por ejemplo a que la clave pública e sea 11.
Nota: para claves mayores que 50 bits, el cálculo de los NNC puede tardar varios minutos y sobre los 60 bits, más de 8 horas.

5.10. ¿Por qué aumenta tanto el tiempo de cómputo de los NNC cuando incrementamos el tamaño del módulo?

5.11. Genera a continuación el log correspondiente de los NNC y observa la distribución de esos números en cada informe.

5.12. De acuerdo a lo observado en el punto 5.10, si se usa RSA con n = 2.048 bits, y con este algoritmo se cifra una clave de sesión del AES de 256 bits, o bien se firma un hash SHA-2 de un documento, ¿es posible que alguno de los NNC de la clave RSA de destino, en el primer caso de cifra, o alguno de los NNC de la clave RSA de emisor, en el segundo, tenga 256 bits? ¿Por qué?

Comprueba tu trabajo:

Clave a) p = 1.013, q = 1.021, e = 7, NNC = 21
0, 1, 46.597, 46.598, 211.716, 258.314, 341.382, 387.979, 387.980, 434.577, 434.578, 599.695, 599.696, 646.293, 646.294, 692.891, 775.959, 822.557, 987.675, 987.676, 1.034.272

Clave b) p = 521, q =1.009, 3 = 11, NNC = 33
0, 1, 12.108, 15.134, 60.540, 63.567, 79.712, 94.847, 159.421, 162.448, 207.854, 210.880, 222.988, 222.989, 235.096, 239.134, 242.161, 283.528, 286.555, 290.593, 302.700, 302.701, 314.809, 317.835, 363.241, 366.268, 430.842, 445.977, 462.122, 465.149, 510.555, 513.581,525.688

Clave c) p = 691, q = 877, e = 7, NNC = 49
0, 1, 22.802, 22.803, 32.731, 45.605, 55.533, 55.534, 78.336, 79.212, 102.014, 102.015, 124.817, 190.026, 212.828, 212.829, 235.630, 235.631, 245.559, 258.433, 268.361, 268.362, 291.164, 291.165, 292.040, 313.967, 314.842, 314.843, 337.645, 337.646, 347.574, 360.448, 370.376, 370.377, 393.178, 393.179, 415.981, 481.190, 503.992, 503.993, 526.795, 527.671, 550.473, 550.474, 560.402, 573.276, 583.204, 583.205, 606.006

Clave d) p = 1.013, q = 1.021, e = 13, NNC = 65
0, 1, 27.396, 31.448, 46.597, 46.598, 76.943, 82.054, 123.541, 150.937, 154.989, 170.139, 182.385, 211.716, 232.991, 237.043, 258.314, 264.438, 274.478, 278.530, 305.926, 311.036, 340.367, 341.382, 387.979, 387.980, 429.467, 434.577, 434.578, 476.155, 491.304, 495.356, 511.520, 522.753, 538.917, 542.969, 558.118, 599.695, 599.696, 604.806, 646.293, 646.294, 692.891, 693.906, 723.237, 728.347, 755.743, 759.795, 769.835, 775.959, 797.230, 801.282, 822.557, 851.888, 864.134, 879.284, 883.336, 910.732, 952.219, 957.330, 987.675, 987.676, 1.002.825, 1.006.877, 1.034.272

Figura 13. Distribución de números pares (rojo) e impares (negro) en los NNC.

10 bits	20 bits	30 bits	40 bits	50 bits
p: 31 q: 29 n: 899 e: 11 d: 611	p: 743 q: 1.021 n: 758.603 e: 11 d: 206.411	p: 24.419 q: 22.751 n: 555.556.669 e: 11 d: 151.502.591	p: 757.579 q: 794.711 n: 602.056.364.669 e: 11 d: 54.732.255.671	p: 20.130.829 q: 29.301.191 n: 589.857.265.517.339 e: 11 d: 160.870.149.841.451
NNC: 33	NNC: 33	NNC: 33	NNC: 33	NNC: 33
0	0	0	0	0
1	1	1	1	1
29	40.866	9.914.115	48.653.238.537	22.686.095.517.457
30	44.579	34.797.076	57.923.732.761	22.686.095.517.458
58	98.818	54.942.750	71.563.185.078	45.372.191.034.915
116	109.221	72.622.107	80.833.679.301	75.143.734.435.500
202	150.086	92.767.781	115.289.887.377	94.742.364.709.227
232	150.087	117.650.742	124.560.381.601	97.829.829.952.958
233	194.666	127.564.856	173.213.620.138	117.428.460.226.685
318	205.810	127.564.857	173.213.620.139	120.515.925.470.416
349	248.905	144.951.183	174.795.445.092	140.114.555.744.143
376	252.619	155.475.772	184.065.939.315	151.211.796.015.365
405	259.308	182.507.607	231.137.352.899	173.897.891.532.823
407	300.173	220.332.638	244.776.805.216	193.496.521.806.549
434	344.753	245.215.599	254.047.299.439	196.583.987.050.281
435	355.897	255.129.713	255.629.124.392	216.182.617.324.007
436	359.611	272.516.040	297.774.001.739	238.868.712.841.465
463	398.992	283.040.629	304.282.362.930	350.988.552.675.874
464	402.706	300.426.956	346.427.240.277	373.674.648.193.332
465	413.850	310.341.070	348.009.065.230	393.273.278.467.058
492	458.430	335.224.031	357.279.559.453	396.360.743.710.790
494	499.295	373.049.062	370.919.011.770	415.959.373.984.516
523	505.984	400.080.897	417.990.425.354	438.645.469.501.974
550	509.698	410.605.486	427.260.919.577	449.742.709.773.196
581	552.793	427.991.812	428.842.744.530	469.341.340.046.923
666	563.937	427.991.813	428.842.744.531	472.428.805.290.654
667	608.516	437.905.927	477.495.983.068	492.027.435.564.381
697	608.517	462.788.888	486.766.477.292	495.114.900.808.112
783	649.382	482.934.562	521.222.685.368	514.713.531.081.839
841	659.785	500.613.919	530.493.179.591	544.485.074.482.424
869	714.024	520.759.593	544.132.631.908	567.171.169.999.881
870	717.737	545.642.554	553.403.126.132	567.171.169.999.882
898	758.602	555.556.668	602.056.364.668	589.857.265.517.338

Figura 14. Informe de los NNC entregado por genRSA v2.1 para claves de 10, 20, 30, 40, y 50 bits, con clave pública e = 11 y 33 Números No Cifrables.

Madrid, 6 de mayo de 2019
Dr. Jorge Ramió Aguirre