Un criterio de optimalidad para primos *strong*

R. Durán Díaz

Instituto de Física Aplicada, CSIC
C/ Serrano, 144, 28006-Madrid
raul@iec.csic.es

J. Muñoz Masqué

Instituto de Física Aplicada, CSIC
C/ Serrano, 144, 28006-Madrid
jaime@iec.csic.es

RESUMEN

Después de repasar los principales protocolos TCP/IP que hacen uso de sistemas criptográficos, se pasa a estudiar más a fondo los requisitos exigibles a los factores primos del módulo de RSA. Se recuerda el concepto de primo *strong* y se propone una función que sirve como criterio de optimalidad para esos primos. Se repasa el algoritmo de Gordon para obtenerlos y se propone uno alternativo para generar primos *strong* óptimos. Se aducen tiempos de computación experimentales para éste último.

1. Primos *strong* y RSA

La amplia difusión que ha alcanzado el uso de Internet para todo tipo de comunicaciones ha despertado la preocupación por garantizar la seguridad y protección de la información que circula por este medio.

Estos requisitos han dado lugar al nacimiento de protocolos que, apoyados en la torre TCP/IP, son criptográficamente seguros. Algunos, como S-HTTP, S/MIME o PEM-MIME trabajan en la capa de aplicación. Más modernamente, SSL o PCT lo hacen ya en la capa de transporte e incluso otros, como IPSec, están diseñados para la capa IP.

Todos estos protocolos (y otros muchos) se fundamentan en un corto número de sistemas criptográficos: DES, para las técnicas de clave secreta y RSA para los esquemas de clave pública. Por ello resulta de capital importancia asegurar la inviolabilidad de esos sistemas. En particular, queremos fijar nuestra atención en el protocolo universalmente utilizado para los esquemas de clave pública: RSA.

Desde la publicación de [1], los creadores de RSA recomendaron el uso de los llamados primos *strong* para los factores del módulo de RSA $n = p \cdot q$, como uno de los métodos de proteger el sistema contra los ataques de algunos poderosos enemigos, tales como [2] ó [3]. En nuestro artículo, proponemos un método que nos permite decidir acerca de la optimalidad de un primo *strong*, pues, como a continuación veremos, su caracterización típica es más bien cualitativa.

2. Caracterización de los primos *strong*

Definición 1. Un número primo p se dice que es *strong* (cf. [4, 4.52]) si verifica: (1) $p - 1$ tiene un factor primo grande r; digamos $p - 1 = ra$; (2) $p + 1$ tiene un factor primo grande s; digamos $p + 1 = sb$; y por último (3) $r - 1$ tiene un factor primo grande t; digamos $r - 1 = tc$.

Introducimos ahora una propiedad de los números enteros que nos va a permitir caracterizar la optimalidad de un primo *strong*. Decimos que un primo *strong* es óptimo si los enteros r, s y t de la *Definición 1* son los más grandes posibles. Por lo tanto, la manera más natural de medir la optimalidad de un primo *strong* es calcular el valor de

$$\sigma(p) = a + b + c = \frac{p-1}{r} + \frac{p+1}{s} + \frac{r-1}{t}.$$

Se verifican las siguientes proposiciones:

Proposición 1. Si p es tal que r, s y t son impares, entonces $\sigma(p) \geq 12$.

Proposición 2. Si p no satisface las condiciones de *Proposición 1*, entonces, o bien p es un primo de Fermat o de Mersenne, o bien se tiene $p = 1 + ra$, siendo r un primo de Fermat y a un entero arbitrario.

Dado un primo p que satisfaga (1) – (3), la suma $a + b + c$ toma el valor mínimo cuando r, s, t son los factores primos mayores de $p - 1$, $p + 1$, $r - 1$, respectivamente. Así pues, sea $S(n)$ el factor primo mayor de n y sea $\sigma: \mathbb{N} - \{1,2\} \to \mathbb{N}$, la función

$$\sigma(n) = \frac{n-1}{S(n-1)} + \frac{n+1}{S(n+1)} + \frac{S(n-1)-1}{S(S(n-1)-1)}.$$

Teorema. Para todo primo $p \geq 23$, se tiene $\sigma(p) \geq 12$. Por tanto un primo *strong* es óptimo si y sólo si $\sigma(p) \geq 12$.

Corolario 1. Un primo *strong* $p > 29$ es óptimo (en adelante, PSO) si y sólo si verifica:

(i) $\frac{p-1}{6}$ es 1-seguro,

(ii) $S(p-1) = \frac{p-1}{6}$,

(iii) $S(p+1) = \frac{p+1}{4}$.

Corolario 2. Obtener PSOs es equivalente a obtener primos 1-seguros r tales que $s = (1+3r)/2$ y $p = 1 + 6r$ sean ambos primos.

3. Distribución de σ

Por razones de espacio no podemos incluir las gráficas de la función $p \to \sigma(p)$ que hemos representado para $3 \le p \le 1200$, ni la de la función $\pi_\sigma : [0,+\infty) \to \mathbb{N}$, $\pi_\sigma(x)$ = número de PSOs menores o iguales que x con $3 \le x \le 25 \cdot 10^6$. A continuación mencionamos dos propiedades muy significativas de tales gráficas.

(a) La acotación $\sigma(p) \ge 12$ es específica de los números primos. De hecho, existen muchos números compuestos n tales que $\sigma(n) < 12$; por ejemplo, si p es un primo 2 veces seguro, (*i.e.*, si $p = 2p_1 + 1$, $p_1 = 2p_2 + 1$, siendo ambos p_1 y p_2 primos), entonces $n = 2p - 1$ verifica $\sigma(n) = 8$. El porcentaje de valores de n tales que $\sigma(n) \le 100$ en el rango $3 \le n \le 1200$ es 86,25%. En cuanto a la función $p \to \sigma(p)$, $3 \le n \le 1200$, el porcentaje correspondiente es 78,97%.

(b) Si bien la gráfica de π_σ muestra que la densidad de PSOs es bastante baja, el perfil creciente que manifiesta dicha curva hace pensar que realmente existen infinitos PSOs.

4. Algoritmo de Gordon

En [5] se presenta un algoritmo que permite obtener PSOs de modo sencillo y cuyo tiempo de ejecución es sólo un 19% más que el necesario para hallar un primo cualquiera. Sin embargo, en ese trabajo no se da ninguna indicación del "grado de optimalidad" de los PSOs así calculados. La función σ es un candidato natural para medir ese grado de optimalidad. Siguiendo el algoritmo de [5], se ha generado la siguiente lista de pares $(p, \sigma(p))$:

(183006137,64708)	(15712927081,1455392)
(5596137293,624016)	(1067010743,122202)
(1932128293,235794)	(261699967,111930)
(1691189693,257722)	(446977967,14560)
(20510989,25456)	(57138093379,6567698)
(112548727,25002)	(3822293537,746820)
(2029062677,316698)	(11969437009,1232828)

Los valores de $\sigma(p)$ computados para cada p muestran claramente que esta técnica —si bien rápida en cuanto a tiempo de ejecución— produce PSOs que distan mucho de ser óptimos, medida la optimalidad mediante la función σ. Obsérvese que, en el ejemplo, el menor valor de $\sigma(p)$ obtenido es 14560, muy lejos del valor mínimo teórico.

5. Obtención de PSOs

Teniendo a la vista lo dicho en el *Corolario 2*, se propone a continuación un algoritmo para obtener PSOs. Los pasos son los siguientes:

1. Se selecciona un entero n aleatoriamente. Se busca el primer número primo r tal que $n \le r$ y además r es un primo 1-seguro.

2. Se calcula $s = (1 + 3r)/2$. Si s no es primo, se vuelve al paso 1.

3. Se calcula $p = 1 + 6r$. Si p no es primo, se vuelve al paso 1. Si lo es, se devuelve p.

Es fácil comprobar que se tiene:

$$\sigma(p) = \frac{p-1}{r} + \frac{p+1}{s} + \frac{r-1}{t} = 12,$$

por lo que realmente el algoritmo devuelve un PSO. El algoritmo se ha implementado en un Pentium II, con una velocidad de reloj de 233 MHz, usando la librería de multiprecisión **gmp** de *GNU* en versión 3.1.

Se han efectuado unas cuantas pruebas numéricas para calibrar el tiempo de ejecución de este algoritmo. En concreto, eligiendo como primer primo *strong* óptimo p = 12163 y calculando los 1370 siguientes, hasta llegar a p = 230445283, se ha observado que el 88% de los tiempos empleados son inferiores a 1 segundo de CPU, si bien existen picos que alcanzan valores de hasta 2,90 segundos. El promedio es de, aproximadamente, 466 milisegundos.

Hemos realizado otros cálculos en el entorno de 10^{30} y los valores medios de cálculo para obtener un primo *strong* utilizando el mismo algoritmo y la misma máquina han sido de 711,504 segundos, es decir, 11 minutos y 51,5 segundos, aproximadamente.

6. REFERENCIAS

[1] R.L. Rivest, A. Shamir, L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, Vol. 21, pp. 120–126, 1978.

[2] J.M. Pollard, "Theorems on factorization and primality testing", Math. Proc. Cambridge Philos. Soc., Vol. 76, pp. 521–528, 1974.

[3] H.C. Williams, "A $p + 1$ method of factoring", Math. Comp. Vol. 39, pp. 225–239, 1982.

[4] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, 1997.

[5] J. Gordon, "Strong Primes are Easy to Find", Advances in Cryptology: Euro Crypto '84, LNCS 209, Springer-Verlag, Berlin, pp. 216–223, 1984.