The Quadratic Sieve

Facultad De Ingeniería, Universidad De Cuenca
CRIPTOLOGIA
Freddy L. Abad L.
{freddy.abadl}@ucuenca.edu.ec

***Abstract* .- Una de las metas en la actualidad ha sido encontrar formas mejores y rápidas de factorizar números compuestos ha sido una meta de los matemáticos desde el principio de los tiempos. La propuesta inicial para realizar esta factorizacion de numeros compuestos ha sido dividir los números por primos grandes hasta obtener la factorización, este método es conocido como Trial Division. Sin embargo, no hubo mejora de este método hasta que Fermat aplicar la factorización por diferencia de dos cuadrados. El método de Fermat es mucho más rápido que Trial Division, sin embargo en el mundo real factorizar un módulo de RSA con cientos de dígitos resulta demasiado lento. Dada este desarrollo se creó nuevos métodos tales como curva elíptica, Pollard, p-1 y p. Algunos de los métodos más rápidos utilizan el mismo truco que Fermat por ejemplo: criba cuadrático, criba de campo numérico, etc. El presente informe pretende resumir de manera clara y concisa el método de la criba cuadrático.**

***Keyword* .- quadratic, sieve, cryptology**

I. INTRODUCCIÓN

Una de las metas en la actualidad ha sido encontrar formas mejores y rápidas de factorizar números compuestos ha sido una meta de los matemáticos desde el principio de los tiempos. La propuesta inicial para realizar esta factorizacion de numeros compuestos ha sido dividir los números por primos grandes hasta obtener la factorización, este método es conocido como Trial Division. Sin embargo, no hubo mejora de este método hasta que Fermat aplicar la factorización por diferencia de dos cuadrados $a^2 - b^2 = (a-b)(a+b)$. El método de Fermat es mucho más rápido que Trial Division, sin embargo en el mundo real factorizar un módulo de RSA con cientos de dígitos resulta demasiado lento. Dada este desarrollo se creó nuevos métodos tales como curva elíptica, Pollard, $p-1$ y p. Algunos de los métodos más rápidos utilizan el mismo truco que Fermat por ejemplo: criba cuadrático, criba de campo numérico, etc. El documento se centrará en el método de criba cuadrático.

II. CRIBA CUADRÁTICO

El criba cuadrático, llamado QS, fue inventado por Carl Pomerance en 1981, a partir de las ideas de Kraitchik y Dixon. El QS fue el algoritmo de factorización más rápido conocido hasta que se presentó el criba de campo numérico en 1993. El QS sigue siendo más rápido para números de hasta 110 dígitos de largo.

III. CÓMO FUNCIONA

Si n es el número a ser factorizado. La QS intenta encontrar dos números x e y tales que $x \equiv \pm y \; mod\,(n)$ y $x^2 \equiv y^2 mod\,(n)$, lo cual implica $(x-y)(x+y) \equiv 0$ donde se calcula $(x-y, n)$ utilizando el algoritmo de euclides para saber si es un divisor trivial. Después hay que definir

$$Q(x) = (x + [\sqrt{b}])^2 - n = \overline{x}^2 - n$$

y determinar

$$Q(x1), Q(x2), ..., Q(xk).$$

Se tiene que para to x

$$Q(x_{i1})Q(x_{i2})...Q(x_{ir}) \equiv (x_{i1}x_{i2}...x_{ir})^2 \; (mod\; n).$$

A. CONFIGURACIÓN DE UNA BASE DE FACTOR Y UN INTERVALO DE CRIBADO

Ahora que se reconoce el esquema básico de QS, es necesario encontrar una manera eficiente de determinar x_i y conseguir un producto de $Q(x_i)$ para obtener un cuadrado. Se sabe que los exponentes de los factores del producto necesitan ser pares. Ahora es necesario factorizar cada $Q(x_i)$, para un conjunto de números primos pequeños llamados base de factores. Para hacer que $Q(x_i)$ sea pequeño, es necesario seleccionar x próximo a 0. Por lo que se establece un M y se considera únicamente a x en el intervalo $[-M, M]$. Ahora si x está en ese intervalo, y alguna p primo que divide a $Q(x)$, entonces:

$$(x-[\sqrt{n}])^2 \equiv n \; (mod\; p)$$

B. CRIBADO

Se toma cada valor de x de la base de factores, calcular el $Q(x_i)$ y comprobar si se factoriza por completo en la base de factores. Si es así tiene suavidad, de lo contrario es descartado y se toma el siguiente elemento.
Este proceso resulta ineficiente ya que se toma un elemento a la vez, para evitar esto se toma un intervalo de cribado.
Cada procesador de la computadora trabajara con un subintervalo diferente. En resumen se obtiene lo siguiente:

$$Q(x) = s^2 \equiv 0 \ (mod\ p), x \in Z_p.$$

Esto puede ser resuelto por el algoritmo Shanks - Tonelli, donde se obtienen dos soluciones: S_{1p} y $S_{2p} = p - S_{1p}$.

Existe algunas maneras para hacer el cribado. Una manera es tomar un subintervalo y poner $Q(x)$ en una matriz para cada x_i en el subintervalo.

Una segunda manera menos exacta, pero es mucho más rápida. En lugar de trabajar con los valores de $Q(x)$ durante algún intervalo, es registrar el número de bits de la $Q(x_i)$ es una matriz. Hay que tomar en cuenta los errores de redondeo y el hecho de que muchos números no están libres de cuadrados.

C. CONSTRUYENDO LA MATRIZ

Si $Q(x)$ hace completamente de factor, entonces se coloca los exponentes $(mod\ 2)$ de los primos en la base del factor en un vector como se describió anteriormente. Todos estos vectores deben estar en la matriz A, por lo que las filas representan la $Q(x_i)$, y las columnas representan los exponentes $(mod\ 2)$ de los primos en la base del factor.

Puede haber varias maneras de obtener un cuadrado perfecto de la $Q(x_i)$, lo cual es bueno, ya que muchos de ellos no nos darán un factor de n. Así que dado $Q(x_1), Q(x_2), ..., Q(x_k)$, entonces hay que encontrar soluciones para:

$$Q(x_1)e_1 + Q(x_2)e_2 + ... + Q(x_k)e_k.$$

donde el e_i sea 0 o 1.
Así que si un $\overline{a_i}$ es la fila correspondiente a $Q(x_i)$, entonces se tiene

$$\overline{a_1}e_1 + \overline{a_2}e_2 + ... + \overline{a_k}e_k \equiv \overline{0}\ mod(2)$$

Esto significa que se tiene que resolver

$$\overline{e}\,A = \overline{0}\ mod(2)$$

Donde

$$\overline{e} = (e_1, e_2, ..., e_k)$$

IV. VARIANTE: EL CRIBA CUADRÁTICO MÚLTIPLE POLINÓMICO (MPQS)

Como su nombre indica, el MPQS utiliza varios polinomios en lugar de Q (x) en el algoritmo, y fue sugerido por primera vez por Peter Montgomery. Estos polinomios son todos de la forma

$$Q(x) = ax^2 + 2bx + c$$

donde se eligen a, b *y* c según ciertos criterios. La motivación de este enfoque es que mediante el uso de varios polinomios, se puede hacer que el intervalo de cribado sea mucho más pequeño, lo que hace $Q(x)$ más pequeño, lo que a su vez significa que una mayor proporción de valores de $Q(x)$ se factoriza por completo sobre la base del factor.

Uno de los problemas conocidos con este método es el costo de conmutación de los polinomios. Pomerance dice que si el costo de la conmutación es aproximadamente $25-30\%$ del costo total, no debería usarse este método. Al cambiar un polinomio, obviamente se necesita nuevos coeficientes, pero para cada nuevo polinomio también se debe resolver $Q(x) \equiv 0\ mod(p)$ para cada primo p en la base de factores, que es la carga más pesada en polinomios de conmutación

V. VARIANTE: THE DOUBLE LARGE PRIME MPQS

Esta versión fue empleada por Lenstra, Manasse y otros, en 1993 y 1994 para factorizar RSA-129 y revelar el mensaje secreto: “Las palabras mágicas son aprensivos ossifrage”. Se esperaba que llevaría 23000 años descifrar este mensaje pero gracias a este método solo tardó 8 meses. Este método considera factorizaciones parciales de la $Q(x_i)$. En el proceso de cribado, nos aferramos a $Q(x)$ y su factorización parcial si tenemos:

$$Q(x) = \prod p_i^{e_i} L,\ L > 1,\ L \leq {p_{max}}^2$$

El factor L debe ser primo de su definición anterior.

VI. LA ELIMINACIÓN GAUSSIANA

Un paso crítico en el proceso de factorización es la etapa de eliminación Gaussiana. La matriz que se forma es enorme, y casi cada entrada es un 0. Esta matriz se llama dispersa. La reducción de esta matriz

utilizando técnicas estándar de álgebra lineal elemental puede acelerar considerablemente. Una consideración trivial es que si tenemos una columna con sólo un 1, podemos eliminar la fila asociada con él. No hay manera posible para que $Q(x)$ sea un factor en un cuadrado. Hay dos algoritmos que hacen la eliminación Gaussiana de una matriz sobre un campo finito: Wiedemann y Lanczos .

VII. TIEMPO DE EJECUCIÓN

Cuando se tiene que el número de primos en la base de factores es pequeña, no es necesario tantas factorizaciones de $Q(x)$ para obtener el posible factor de n. El problema sigue siendo encontrar cada uno de las factorizaciones, si tuviéramos una enorme lista de primos, nuestro problema sería obtener todos esos números para crear una matriz lo suficientemente grande para reducirla. Por lo tanto, el número de primos debe establecerse para optimizar el rendimiento. El valor óptimo para el tamaño de la base de factores es aproximadamente:

$$B = \left(e^{\sqrt{ln(n)ln(ln(n))}} \right)^{\sqrt{2}/4}$$

El intervalo del cribado es :

$$M = \left(e^{\sqrt{ln(n)ln(ln(n))}} \right)^{3\sqrt{2}/4}$$

Su tiempo de funcionamiento asintótico para QS es:

$$O\left(e^{\sqrt{ln(n)ln(ln(n))}} \right)$$

El criba de campo numérico, por comparision, que es el algoritmo de factorización más rápido conocido públicamente, tiene tiempo de ejecución

$$O\left(e^{1.9223((ln(n)^{1/3}(ln(ln(n)))^{2/3})} \right)$$

VIII. RSA

La seguridad de RSA se basa en la dificultad de factorizar enteros. La factorización es un éxito con un módulo RSA de 129 dígitos. Los módulos RSA de 512 bits, o unos 155 dígitos serían factibles de factorizar, y de hecho se han factorizado. En agosto de 1999, un equipo que incluye a Arjen Lenstra y Peter Montgomery factorizaron un módulo RSA de 512 bits utilizando el criba numérico de campo en 8400 MIPS años (8,4 mil millones instrucciones por segundo años).

Las estimaciones actuales dicen que un módulo de 768 bits será bueno hasta 2004, así para uso a corto plazo o personal, tal tamaño de clave es adecuado. Sin embargo, para el *uso corporativo, se sugiere un módulo de 1024 bits, y se sugiere un módulo de bit 2048 para un uso mucho más permanente.*

Estas sugerencias tienen en cuenta los posibles avances en las técnicas de factorización y los aumentos de velocidad del procesador.

Riesel muestra que es posible crear un algoritmo para factorizar enteros en tiempo casi polinómico, por lo que ciertamente hay espacio para mejoras. Sin embargo, si un ordenador cuántico se construye alguna vez con un número suficiente de qubits, Peter Shor ha descubierto un algoritmo para factorizar enteros en tiempo Polinómico en él.

Entonces RSA debe ser retirado y sustituido por otros esquemas de cifrado como los módulos requeridos para ser seguro sería mucho más grande que lo que sería conveniente.

REFERENCIAS

[1] Landquist, E. (2001). The Quadratic Sieve Factoring Algorithm. MATH 488: Cryptographic Algorithms.

[2] Editores Wikipedia. (2019). Criba Cuadrática. Online available: https://es.wikipedia.org/wiki/Criba_cuadr%C3%A1tica

[3] Leonel Sergio Carrasco Perez. (2019). FACTORIZACIÓN DE ENTEROS. Online available: http://mat.izt.uam.mx/mcmai/documentos/tesis/Gen.09-O/Carrasco-LS-Tesis.pdf

[4] Editores Unionpedia. (2019). Criba Cuadrática. Online available: https://es.unionpedia.org/i/Criba_cuadr%C3%A1tica