CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL
INSTITUTO POLITECNICO NACIONAL

DEPARTAMENTO DE INGENIERIA ELECTRICA
SECCION DE COMPUTACION

"FACTORIZACION DE NUMEROS ENTEROS Y SU APLICACION A LA CRIPTOGRAFIA"

T E S I S

QUE PRESENTA EL LIC.

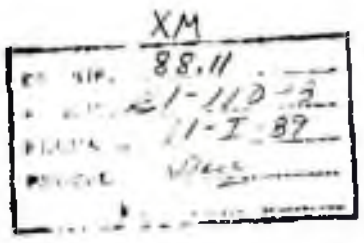
FERNANDO VAZQUEZ GARCIA

PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS
EN LA ESPECIALIDAD DE
INGENIERIA ELECTRICA
CON OPCION DE COMPUTACION

TRABAJO DIRIGIDO POR EL DR.

GUILLERMO MORALES LUNA

BECARIO DE COSNET. MEXICO D. F. A 18 DE OCTUBRE DE 1988


XM
88.11
21-110-2
11-I-87

A mis padres :

Rafael Vázquez Huerta

María de la Luz García de Vázquez

con un profundo amor y respeto.

Agradecimientos

Mi admiración y respeto al Dr. Guillermo B. Morales Luna por la oportunidad que me brindo de trabajar con él en esta tesis.

Mi más sincero agradecimiento a los Doctores Manuel E. Guzmán Rentería y Horacio Tapía Recillas por su tiempo dedicado a la lectura de esta tesis.

Al Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, por permitirme realizar mis estudios en su Sección de Computación.

A COSNET por la beca otorgada a mi persona, que hizo posible la realización de mis estudios.

FACTORIZACION DE NUMEROS ENTEROS Y SU APLICACION A LA CRIPTOGRAFIA

Introducción p. 2

1. PRIMICIDAD Y FACTORIZACION DE NUMEROS ENTEROS 5
   1.1. Primicidad 5
   1.2. Factorización 14
      1.2.1. Algortimos para factorización 16

2. CRIPTOGRAFIA 24
   2.1. Importancia de la Criptografía 24
   2.2. Técnicas de encriptamiento y desencriptamiento de mensajes 27
   2.3. Nuevo esquema en Criptografía 32

3. APLICACIONES 39
   3.1. Algoritmos para la generación de números primos 39
   3.2. Sistema Criptográfico RSA 48

Conclusiones 54

Apéndice 55

Bibliografía 59

Introducción

La teoría de números es considerada como la reina de las matemáticas, en parte por su intrínseca belleza en la demostración de sus teoremas, que hasta en cierto punto hacen de su estudio una contemplación a la abstracción pura y la otra por la potencia práctica de sus consecuencias. Existen muy antiguos problemas en esta rama que no han logrado resolverse, los más interesantes son, dado un número entero **N** mayor a la unidad, ¿ cómo determinar si éste es un número primo? y en dado caso de no serlo, ¿ cómo determinar sus factores primos ?; éstos han merecido la atención de los grandes matemáticos, desde los griegos hasta nuestra época, logrando en cada una de ellas avances que han permitido vislumbrar ya sea, la posible existencia de un algoritmo de complejidad polinomial para determinar si el número **N** es primo o no, así como la posible no existencia de un algoritmo de igual complejidad para la factorización de números enteros. En el presente trabajo se expone de una forma somera, los algoritmos con que se cuenta para dar solución a los problemas ya mencionados.

Difícil resulta imaginar la existencia de alguna relación entre los números primos y la llamada teoría de la información y mucho más difícil resulta imaginar aplicación alguna entre ellas. En la teoría de la información se tiene un problema bastante serio, que es el de cómo poder proteger la información, cuando ésta es transmitida por algún medio en el cuál cualquier persona puede tener acceso a ella?, el poder conocer información clasificada como secreta, puede llegar a tener un gran costo. Se han propuesto varias técnicas para resolver este problema, pero la mayoría de ellas han sucumbido ante el desarrollo de la computación; todas estas técnicas se encuentran enmarcadas en lo que se ha llamado **CRIPTOGRAFIA**.

En el año de 1976, **DIFFIE** y **HELLMAN** propusieron una nueva dirección que podría tomar la Criptografía, basada en funciones de

UN SOLO SENTIDO. Una función f es de un solo sentido si es fácil de computar, más no asi su inversa. esto es, la evaluación de f con sus parámetros es rápida, sin embargo su inversa sin el conocimiento de sus parámetros no es factible calcularla y una vez conocidos éstos, es sencilla de evaluar. Las anteriores funciones pueden ser encontradas en problemas $\mathcal{NP}$, un problema P es $\mathcal{NP}$, si en un tiempo polinomial se puede decidir si dada una posible solución a P, ésta es o no una solución efectiva. El problema P es *$\mathcal{NP}$-completo*, si P es $\mathcal{NP}$ y si para cualquier otro problema $\mathcal{NP}$ (digamos Q), Q es reducible a P en tiempo polinomial. Entre los problemas *$\mathcal{NP}$-completos* están :

(i) Isomorfismo entre subgráficas; dadas dos gráficas G_1 y G_2, ¿es G_1 isomorfa a una subgráfica de G_2 ?.

(ii) Ciclo Hamiltoniano ó problema del agente viajero; dada una gráfica G, ¿existe en G un ciclo que pase solamente una vez por cada vértice?.

(iii) Suma de subconjuntos; dados $n_1, \ldots, n_m$ números enteros y una suma S, ¿qué subconjunto de los números n_i da como suma S ?.

(iv) Máxima subgráfica planar; dada una gráfica G, ¿contiene G una subgráfica planar (una gráfica se dice planar, si ésta puede dibujarse en el plano de tal forma que sus arcos solamente se intersecten en los vértices) con al menos k vértices?.

En general los sistemas criptográficos de llave pública (ver pág. 32) consideran instancias de problemas *$\mathcal{NP}$-completos*, en los cuales tanto el encriptamiento como el desencriptamiento de mensajes, poseen llaves y cada una de ellas un algoritmo para calcularlas, si uno ignora dicho algoritmo, en partícular para querer desencriptar un mensaje le llevará el tener que enfrentarse a calcular todas las posibles soluciones del problema *$\mathcal{NP}$-completo*.

Es un problema abierto en matemáticas el demostrar la existencia o no de un algoritmo de tiempo polinomial para la factorización de números enteros, de acuerdo con las últimas investigaciones en este

campo, se conjetura que dicho problema es $\mathcal{NP}$. Es en el año de 1978 cuando **RIVEST, SHAMIR** y **ADLEMAN**, proponen un sistema criptográfico con la orientación de Diffie y Hellman, el cuál se basa en la infactibilidad de querer factorizar un número entero **N** de más de 120 dígitos; es aquí donde entran los números primos, en su sistema **N** es el producto de dos grandes números primos, con ellos es posible proteger la información que se quiera trasmitir.

En este trabajo también se exponen algunas técnicas para la protección de la información, así como una implementación del sistema criptográfico de Rivest, Shamir y Adleman, en el cuál se muestra una hermosa aplicación de la teoría de números a la teoría de la información.

1. PRIMICIDAD Y FACTORIZACION DE NUMEROS ENTEROS

Lo que a continuación se presenta, es un resumen general de los algoritmos más importantes para la prueba de primicidad y factorización de números enteros; cabe advertir que no son los unicos, pero si los más utilizados y/o que han permitido diseñar algoritmos más eficientes.

El objetivo principal es mostrar las características más importantes de cada algoritmo que se considere, no un análisis exahustivo; análisis que en muchos casos no ha sido del todo completo por la enorme complejidad que presenta. Es por lo anterior que el lector notará inmediatamente la ausencia de un uso profundo de lenguaje matemático en el desarrollo del trabajo; si el lector desea adentrarse más al tema, puede consultar la bibliografía dada.

De aquí y en lo que resta del trabajo, **N** denotará un número entero mayor que la unidad.

1.1. Primicidad

Uno de los problemas de gran interés y de mayor antiguedad en la teoría de números, es el que dado un número impar **N**, se desea saber si éste es un número primo o un número compuesto; para ello, **N** debe ser sometido a una serie de pruebas de diversa indole; a tales pruebas se les llama "**pruebas de primicidad**".

A continuación se exponen algunos resultados en la teoría de números, que han permitido desarrollar algoritmos para la prueba de primicidad factibles a ser implementados en un computador

Los primeros intentos en este campo datan de tiempos lejanos, en particular, la "**Criba de Eratóstenes**" es en cierta forma un algoritmo para determinar si **N** es o no un número primo, aunque claro, si **N** es de tamaño pequeño. Los primeros estudios teóricos y por tanto las

direcciones de aplicación para pruebas de primicidad, fueron dadas por el matemático fránces Pierre Fermat en *1640*, el teorema que enunció y demostró es conocido en la actualidad como el "Pequeño Teorema de Fermat", de hecho muchos de los más recientes algoritmos ideados para esta prueba, se basan en dicho teorema y sus generalizaciones.

Teorema 1.1. ("Pequeño Teorema de Fermat")

Si **N** es un número primo y **a** es cualquier otro número entero, entonces se cumple :

$$a^N \equiv a \bmod N,$$

es decir,

$$a^{N-1} \equiv 1 \bmod N .$$

El teorema permite concluir que si $a^N - a$ no es un múltiplo de **N**, para alguna **a**, entonces **N** es un número compuesto. A simple vista podria pensarse que el teorema 1.1 da una pronta solución a la prueba de primicidad, con él podemos saber si **N** es un número compuesto; pero, se puede concluir que si $a^N - a$ es un múltiplo de **N**, entonces **N** es un número primo ?, esto es, es cierto el inverso del teorema 1.1? por ejemplo, 2^2-2 es múltiplo de 2, 2^3-2 es múltiplo de 3, 2^5-2 es múltiplo de 5, etc.; 2, 3, 5 son números primos; esta clase de ejemplos hizo que por algún tiempo se pensara como cierto el inverso del teorema 1.1., hasta que en *1819* el matemático fránces Pierre Fréderic Sarrus mostró que el número $2^{341}-2$ es un múltiplo del número compuesto $341 = 11 \cdot 31$; para ver esto, se tienen los siguientes resultados.

i) Si $a \equiv b \bmod m_1$, ... , $a \equiv b \bmod m_k$, es claro que $a-b$ debe ser un múltiplo de m_i $\forall i$, en partícular lo es también del mínimo común múltiplo de ellos.

ii) Si $a \equiv a' \bmod m$ y $b \equiv b' \bmod m$, entonces $a \cdot b \equiv a' \cdot b' \bmod m$, esto es cierto ya que $a \equiv a' \bmod m \Rightarrow a - a' = k_1 \cdot m \Rightarrow a = a' + k_1 \cdot m$ y $b \equiv b' \bmod m \Rightarrow b - b' = k_2 \cdot m \Rightarrow b = b' + k_2 \cdot m$, luego se tiene,

$a \cdot b = a' \cdot b' + m(a' \cdot k_2 + b' \cdot k_1 + k_1 \cdot k_2) \Rightarrow a \cdot b$ es un múltiplo de m y por tanto $a \cdot b \equiv a' \cdot b' \bmod m$.

Haciendo uso de lo anterior, puede demostrarse que si $a \equiv b \bmod m$ y n es un entero positivo, entonces $a^n \equiv b^n \bmod m$.

Volviendo al contraejemplo del inverso del "pequeño teorema de Fermat", tenemos :

Es claro que $2^5 \equiv 1 \bmod 31$, ahora, $N=11$ es un número primo y por el teorema 1.1. se cumple $2^{11} \equiv 2 \bmod 11$, es decir, $2^{10} \equiv 1 \bmod 11$.

De $2^5 \equiv 1 \bmod 31$ y ii) se tiene $2^{10} = (2^5)^2 \equiv 1^2 \bmod 31$, por tanto, $2^{10} \equiv 1 \bmod 31$ y $2^{10} \equiv 1 \bmod 11$, puesto que $(11,31)=1$, el mínimo común múltiplo de ellos, es su producto, $11 \cdot 31 = 341$; y por i) obtenemos :

$2^{10} \equiv 1 \bmod 341$; aplicando nuevamente ii)
$(2^{10})^{34} \equiv 1^{34} \bmod 341$
$2^{340} \equiv 1 \bmod 341$; es decir
$2^{341} \equiv 2 \bmod 341$.

Pero aún, aunque fuera cierto el inverso del teorema 1.1., computacionalmente, no sería factible revisar todos los enteros a mod N, con números N de tamaño moderado.

Poco a poco se han ido desarrollando pruebas de primicidad para cualquier número N; éstas se han basado en resultados obtenidos para números N que tienen una forma especial; por ejemplo, el matemático suizo L. Euler mostró, que si N es un número de la forma $4 \cdot k+1$, que pueda ser escrito de manera única como la suma de los cuadrados de dos números primos relativos, entonces N es un número primo, el resultado es una consecuencia inmediata del siguiente teorema.

Teorema 1.2.

Sea $N=4k+1$, el cuál puede ser escrito como la suma de los

cuadrados de dos números primos relativos. Entonces N es un número compuesto si y sólo si N puede ser escrito como la suma de los cuadrados de dos números, en dos formas distintas.

Ejemplo 1.1.

Sea $N=4(25300)+1 = 101201$; puede verse que $N=151^2+280^2$, siendo $(151,280)=1$, de esta forma se tiene una representación de N; si existiera otra forma de representar a $N=a^2+b^2$, a y b deben ser menores a $\sqrt{N}$; en este caso es facil determinar a tales números (en caso de existir) a partir de 151, con algunos cálculos se puede encontrar que $N=176^2+285^2$, así tenemos dos representaciones diferentes de N y por el teorema anterior, N es un número compuesto.

Nótese que esto puede ser utilizado para demostrar la primicidad de N, siempre y cuando su representación pueda ser completamente determinada. En el año *1876*, el matemático fránces Edovard A. Lucas enunció y demostró un teorema que ha resultado de gran importancia en el desarrollo de la prueba de primicidad.

Teorema 1.3.

Sea N un número tal que $N-1=\prod_{i=1}^{k} q_i^{\alpha_i}$ es su representación canónica, con q_i distintos números primos y $\alpha_i \geq 1$ $\forall i$, $i=1,\ldots,k$.

Si un entero a puede ser encontrado tal que, $a^{(N-1)/q_i} \not\equiv 1 \bmod N$ y además $a^{N-1} \equiv 1 \bmod N$, entonces N es un número primo.

Se darán algunos hechos en la teoría de números que permitirán bosquejar argumentos y que también, darán validez al teorema.

Denotemos por $\varphi(N)$ la función de Euler, si N es un número primo, entonces $\varphi(N)=N-1$, en otro caso $\varphi(N)<N-1$, por otra parte, la congruencia $a\equiv b \bmod N$, define una transformación de $\mathbb{N}$ a $\{0, 1, \ldots, N-1\}$, la cual es un homomorfismo. Los enteros 0, 1, ..., N-1, representan lo que es llamado las "N diferentes clases residuales módulo N"; si N es un número primo, entonces todas las clases

residuales módulo N forman un CAMPO, además las N-1 clases residuales que no son congruentes a 0 módulo N forman con el producto usual como operación un grupo abeliano de orden N-1, éste último será denotado por M_N.

Si se descartan todos los múltiplos de N, es decir, se consideran solamente todos los números a tales que (a,N)=1, se forma un nuevo tipo de clases residuales llamadas "**CLASES RESIDUALES PRIMITIVAS a módulo N**", estas clases forman un grupo multiplicativo M_N de orden $\varphi(N)$. F. Gauss mostró que si N es un primo, M_N es un grupo cíclico, esto es, existe un elemento cuyo orden es N-1 (pues $\varphi(N)=N-1$) y genera al grupo.

Bajo estos resultados, puede verse que, las condiciones pedidas al entero a en el teorema 1.3, aseguran que éste sea el elemento de orden N-1 del grupo M_N de clases residuales primitivas módulo N y esto por lo demostrado por Gauss, garantiza la primicidad de N.

En su demostración, Lucas no da indicación alguna de cómo poder determinar al número a, no obstante, existe la posibilidad de encontrar a por medio de un búsqueda aleatoria. Es con este tipo de búsqueda que la prueba de primicidad de Lucas es un método de Monte-Carlo.

Una vez determinado dicho número, se tiene una prueba rigurosa de la primicidad de N; no obstante, se tienen dos problemas, primero determinar a; segundo, la representación canónica del número N-1 debe ser conocida y esto en otras palabras es el poder factorizar a N-1, problema que como se verá en la siguiente sección, es bastante más difícil que el de que aquí nos ocupamos. Por esto último la prueba de Lucas generalmente no es factible, a menos que N-1 pueda ser factorizado en forma rápida.

Se define un número de Fermat como $F_n=2^{2^n}+1$, con $n \in \mathbb{N}$.

El matemático P. Pepin demostró el siguiente teorema en *1877*.

Teorema 1.4.

El número de Fermat F_n, es un número primo si y sólo si $3^{(F_n-1)/2} \equiv -1 \bmod F_n$.

Por el teorema 1.3., se debe determinar un número a tal que :

$$a^{(F_n-1)/2} = a^{2^{2^n}-1} \not\equiv 1 \bmod F_n \quad y$$

$$a^{F_n-1} = a^{2^{2^n}} \equiv 1 \bmod F_n$$

hagamos el cambio $x=a^{2^{2^n}-1}$, así $x \not\equiv 1 \bmod F_n$ y $x^2 \equiv 1 \bmod F_n$, obteniendo la congruencia $x^2-1 \equiv (x+1)(x-1) \equiv 0 \bmod F_n$, cuyas dos únicas soluciones son, $x \equiv -1 \bmod F_n$ y $x \equiv 1 \bmod F_n$, ésta última no cumple con la condición $x \not\equiv 1 \bmod F_n$ pedida en el teorema 1.3., por tanto la solución que sirve en este caso es la primera, así se hace necesario determinar un entero a que satisfaga la congruencia $x \equiv a^{(F_n-1)/2} \equiv -1 \bmod F_n$ cuando F_n sea un número primo.

La solución a esta última congruencia recae en la teoría de los residuos cuadráticos, en ella Euler dió un criterio, que en este caso dice que a debe ser un no-residuo cuadrático de F_n, puede verse, que el número 3 es un no-residuo cuadrático de todos los primos de la forma $12 \cdot n \pm 5$, cabe señalar que los números de Fermat cumplen $F_n \equiv 5 \bmod 12$. Lo cual puede observarse de que, $2^{2^1}=4$, $2^{2^2}=16$, $2^{2^3}=256$ son congruentes 4 módulo 12 y por tanto los números de Fermat serán congruentes 5 módulo 12, así F_n un número primo si y sólo si $3^{(F_n-1)/2} \equiv -1 \bmod F_n$.

Nótese que con el anterior teorema se logra un avance en la prueba de primicidad, ya que la representación canónica de N-1 no es requerida, no obstante, la forma de N debe ser particular.

No es sino hasta principios del presente siglo, cuando se logra una prueba de primicidad para números **N** que no tengan forma especial alguna, sin embargo se pide una factorización de una parte de **N-1**.

Teorema 1.6.

Supóngase que $N-1=F\cdot R$, con **F** un número cuya representación canónica es conocida, esto es, $F=\prod_{i=1}^{k} q_i^{\beta_i}$, además $F>R$ y $(F,R)=1$. Si existe un entero **a**, tal que

$$(a^{(N-1)/q_i}-1,N)=1 \quad \forall i=1,\ldots,k$$

y

$$a^{N-1}\equiv 1 \bmod N$$

entonces **N** es un número primo.

La demostración del teorema puede bosquejarse de la siguiente forma.

Sea **p** un posible factor primo de **N**, la primera y segunda condición del teorema 1.6. implican respectivamente que $(a^{(N-1)/q_i}-1,p)=1$ $\forall i$ y $a^{N-1}\equiv 1 \bmod p$. Se ha mencionado que si **p** es un número primo, la congruencia módulo **p** forma un grupo, que se denota por M_p y de orden $p-1$.

Sea **a** un elemento de M_p de orden **d**, se sabe que el orden de un elemento divide al orden del grupo al que pertenece, por tanto $d|p-1$, $d|N-1$ y $d\nmid (N-1)/q_i$ $\forall i$, es decir, se tiene $d|R\cdot F = d|R\prod q_i^{\beta_i}$ y $d\nmid R\, q_j^{\beta_j-1}\prod_{i\neq j} q_i^{\beta_i}$; pero ya que $(R,\prod q_i^{\beta_i})=1$, $d|R\prod q_i^{\beta_i}$ solo puede pasar si $q_i^{\beta_i}|d$ $\forall i$; esto implica que $F|d$ y puesto que $d|p-1$ entonces $F|p-1$, de esta forma el valor más pequeño que puede tomar **p** es igual a $F+1$ y puesto que $F>R$, entonces $p=F+1>\sqrt{N}$. De aquí se ve que **p** no puede existir, por tanto **N** debe ser un número primo.

Existen muchos otros más teoremas que permiten probar la primicidad de números enteros.

En el año de *1983*, apareció en el volumen 117 de ANNALS OF MATHEMATICS un artículo por M. Adleman, C. Pomerance y R. Rumely (APR) en el cual dan a conocer un algoritmo para la prueba de primicidad, éste, cambió radicalmente la eficiencia de dicha prueba para grandes números N carentes de algura forma especial.

La implementación de la prueba tal y como fué expuesta originalmente, la realizó W. G. Dubuque en una máquina DEC KL-10 en el MIT, dicha implementación logró probar la primicidad de un número de 62 digitos en aproximadamente 6 horas. El algoritmo tanto en la parte teórica como en la parte computacional es muy compleja. A continuación se explica en forma general la idea básica del algoritmo.

Se trabaja en campos ciclotómicos, en primer término se lleva a cabo una serie de pruebas para seudoprimos sobre el número N, si cualquiera de éstas falla, entonces se declara a N como un número compuesto; si resultan "exitosas", la información generada por ellas es utilizada para construir una criba restringida de todos los posibles divisores primos de N. Lo anterior se lleva a cabo construyendo un pequeño conjunto $\mathfrak{Z} = \mathfrak{Z}(N)$ de primos.

Se define a un **Primo Euclidiano** con respecto a $\mathfrak{Z}$, como un primo p, tal que todo factor primo de p-1 está en $\mathfrak{Z}$.

En el algoritmo de APR, el conjunto $\mathfrak{Z}$ debe ser lo más pequeño posible, pero de tal forma que el producto de primos Euclidianos con respecto a $\mathfrak{Z}$ sea mayor a $\sqrt{N}$, lo anterior, con el fin de que el tiempo de ejecución del algoritmo sea polinomial en el producto de los elementos que formen a $\mathfrak{Z}$; en el artículo se demuestra que $\mathfrak{Z}$ puede ser formado de tal manera que :

$$\prod_{q \in \mathfrak{Z}(N)} q \; < \; (\log N)^{c \log\log\log N}$$

para todo $N > 100$, con c una constante positiva y calculable.

Dos números auxiliares son construidos, el número inicial I y el número Euclidiano E, éste último es el producto de todos los numeros primos p_i tales que $p_i - 1$ son los factores del número I, estos primos serán Euclidianos respecto $\mathfrak{I}$.

Por ejemplo

Sea $\mathfrak{I}=(2, 3, 5, 7)$ y $I = \prod_{q \in \mathfrak{I}} q = 210$

Tomemos los primos

p_i	2	3	5	7	11	13	17	19	23	29	31	. . .	211
$p_i - 1$	1	2	4	6	10	12	16	18	22	28	30	. . .	210

Los $p_i - 1$ que dividen a I son 1,2,6,10,30,42,70 y 210; por tanto, $E = 2 \cdot 3 \cdot 7 \cdot 11 \cdot 31 \cdot 43 \cdot 71 \cdot 211 = 9\ 225\ 988\ 926$; nótese que todo factor primo de $p_i - 1$ (con p_i en el producto de E) esta en $\mathfrak{I}$; por ejemplo p=71, $p-1 = 70 = 2 \cdot 5 \cdot 7$, con 2, 5, 7 elementos de $\mathfrak{I}$.

Una vez que tanto I como E han sido construidos, para cada par de primos p y q, con p un factor de E y q un factor de p-1, se llevan a cabo pruebas similares a las de Fermat. Si alguna de las pruebas falla para con p y q, entonces N es declarado un número compuesto, en otro caso, aún no es posible declarar a N como un número primo, pero, el número de posibles factores de N que restan por probar es pequeño.

Adleman y Rumely mostraron que cualquier número compuesto que pasa todas las pruebas del tipo de Fermat, a lo más tiene factores primos que unidos forman un conjunto cuya cardinalidad es I. Por otra parte, Lenstra mostró que los números en dicho conjunto son iguales a los residuos de las potencias $N, N^2, N^3, \ldots, N^I \bmod E$ y con una simple división, cualquiera de estos números que sea diferente de 1 y N y que logre dividir exactamente a éste último, será suficiente para declarar a N como número compuesto, en otro caso como número primo.

El tiempo de ejecución se aproxima al tiempo polinomial, para ser más precisos, su tiempo máximo es de :

$$(\log N)^{c \log\log\log N}$$

con c una constante positiva.

Recientemente H. Cohen y W. Lenstra Jr. [4], implementaron el algoritmo de APR con algunas modificaciones, logrando con ello una eficiencia sorprendente, ya que, han podido verificar la primicidad de un número de 100 digitos en aproximadamente 40 segundos. Una diferencia notable de la implementación es que APR trabajan con sumas Gaussianas mientras que Cohen y Lenstra lo hacen con sumas de Jacobi, además, lograron generalizar algunos resultados que permiten multiplicar factores primos, logrando con ello una mayor rapidez de ejecución.

Como resultado del trabajo de Cohen y Lenstra, se obtiene un programa (en algún lenguaje de alto nivel) para la prueba de primicidad muy eficiente, el cual esta basado en un algoritmo que tanto teórica como prácticamente es más avanzado que cualquier otro más reciente.

1.2. Factorización.

El problema de factorización, puede ser enunciado de la siguiente forma :

Dado **N**, determinar su representación canónica.

El problema en sí, consta de dos partes, la primera es determinar si **N** es un número primo o un número compuesto, la prueba de primicidad fue tratada en la anterior sección; ahora bien, si **N** resulta ser un

número compuesto, se presenta el problema de determinar todos sus factores primos. En la actualidad, se cuenta con varios algoritmos para dar solución a este problema por medio de un computador, entre los más importantes se pueden citar : búsqueda directa, diferencia de cuadrados, métodos de John Pollard, método de H. C. Williams y el algoritmo (que es hasta hoy el más rápido) de Richard Schroppel, algoritmo que aún no ha sido publicado, puede factorizar un número N en aproximadamente :

$$C \ln N^{\sqrt{\ln(N)\ln\ln(N)}}$$

pasos.

La siguiente tabla tomada de [2] da el número de operaciones y el tiempo que emplearía tal algoritmo para factorizar un número compuesto N.

dígitos de N	número de operaciones	tiempo
50	1.4×10^{10}	3.9 hrs.
75	9.0×10^{11}	104 días
100	2.3×10^{13}	74 años
200	1.2×10^{28}	3.8×10^{4} "
300	1.5×10^{39}	4.9×10^{15} "
500	1.3×10^{59}	4.2×10^{25} "

La estimación del tiempo está basada en que, cada operación requiere de un microsegundo.

Por otra parte, es conveniente señalar que no siempre es posible obtener una factorización completa de N utilizando exclusivamente un solo método, es necesario el uso de dos o más de ellos.

1.2.1. Algoritmos para factorizacion.

a) BUSQUEDA DIRECTA.

Puede considerársele como el método de "divide y vencerás", aunque lo más acertado sería "más divide que vencerás". Consiste en generar en orden creciente una sucesión de posibles divisores de **N**, éstos deberán estar acotados por $\sqrt{N}$ y son probados para confirmar si dividen exactamente a **N**, en caso de que alguno lo haga, se habrá encontrado un factor de él. Este método es viable cuando **N** es pequeño y/o cuando tiene factores pequeños.

Considérese a $N=2^{109}-1$ [27], con tal número, al método de búsqueda directa le llevaría en un computador, alrededor de 35×10^9 años el encontrar el factor 61654440233248340616559 de **N**.

b) METODO DE FERMAT.

Este es un método sistemático; la idea de Fermat es el de tratar de escribir a un número impar **N** como la diferencia de los cuadrados de dos números, esto es, $N=x^2-y^2=(x+y)(x-y)$ y con ello se obtiene la factorización deseada, he aquí el procedimiento.

Tómese $m=[\sqrt{N}]+1$, donde [ ] significa el mayor entero menor o igual, en caso de que $\sqrt{N}$ sea exacta, se tendrá la representación $N=x^2-0^2$, en otro caso, hacemos $z=m^2-N$ y se verifica si **z** es un cuadrado perfecto, si lo es encontramos $N=x^2-y^2$, con $y^2=z$; en otro caso se intenta con **m+1**, esto es, $z=(m+1)^2-N=m^2+2m+1-N=m^2-N+2m+1=z+2m+1$ y de nueva cuenta se verifica si **z** es un cuadrado perfecto, si no lo es, el procedimiento continua aumentando **m** una unidad más.

Por ejemplo, sea N=23533, se tiene $m=[\sqrt{N}]+1=154$, $z=m^2-N=183$.

siguiendo el procedimiento en forma iterativa, obtenemos los siguientes valores :

m	2m+1	z
154	309	183
155	311	492
156	313	803
157	315	1116
158	317	1431
159	319	1748
160	321	2067
161	323	2388
162	325	2711
163	327	3036
164	329	3363
165	331	3692
166	333	4023
167	335	4356

Se tiene $z=4356=66^2=y^2$, así $x=\sqrt{N+y^2}=\sqrt{27889}=167$; por tanto :

$$N=(x+y)(x-y)=(167-66)(167+66)=101\cdot 233.$$

El método de Fermat es práctico cuando los dos factores son de un valor cercano a $\sqrt{N}$, en otro caso es impráctico.

c) **METODO DE EULER.**

Este método es especial, puesto que sólo es aplicable a enteros que puedan ser escritos como $N=a^2+c\cdot b^2$ en dos formas diferentes, con c un entero común a ellas. El método se basa en la siguiente identidad dada por Lagrange :

$$(x^2+c\cdot y^2)(u^2+c\cdot v^2)=\begin{cases}(x\cdot u+c\cdot y\cdot r)^2+(y\cdot u-x\cdot v)^2\\(x\cdot u-c\cdot y\cdot r)^2+(y\cdot u+x\cdot v)^2\end{cases}$$

la cual puede ser verificada.

Nótese que se puede concluir que el producto de dos enteros de la forma $a^2+c\cdot b^2$, es un entero de la misma forma y con dos diferentes formas de escribirse, a saber $r^2+c\cdot s^2$.

Euler mostró que si N puede ser expresado de dos formas distintas, por ejemplo, $N=a^2+C\cdot b^2$ y $N=c^2+C\cdot d^2$, con $(b\cdot d,N)=1$, entonces N puede ser escrito como el producto de dos números de la forma $r^2+C\cdot s^2$ y los factores de N estan dados por $(N,a\cdot d-b\cdot c)$ y $(N,a\cdot d+b\cdot c)$.

Puede hacerse uso de un algoritmo semejante al de Fermat para encontrar la segunda representación de N, una vez determinada ésta es sencillo calcular los factores de N; ilustremos el método de Euler como sigue :

En el ejemplo 1.1, se mostró que N=101201 es un número compuesto, además N tiene las siguientes representaciones diferentes, $N=151^2+280^2$ y $N=176^2+265^2$, puede comprobarse que $N=151^2+25\cdot 56^2$ y $N=176^2+25\cdot 53^2$, además $(N,53\cdot 56)=1$, satisfaciendo las condiciones pedidas para determinar los factores de N, así $(N,151\cdot 53-56\cdot 176)=17$ y $(N,151\cdot 53+56\cdot 176)=5953$, por tanto $N=17\cdot 5953=101201$.

d) METODOS DE POLLARD.

A mediados de los años 70's, John Pollard dio a conocer dos métodos, que han significado un gran avance en el problema de factorización.

d.1) METODO "p-1".

El primero de ellos en 1974, conocido como el método "p-1", puede en ocasiones, determinar un factor primo p de N muy rápidamente, si es que p-1 es el producto de pequeños factores primos, es decir :

$$p-1=\prod_{i=1}^{k} q_i^{\alpha_i} \qquad \text{con } q_i \text{ números primos y } \alpha_i \in \mathbb{N},\ \forall i$$

La idea del método de Pollard, es el extraer información de un elemento a del grupo M_N. Supóngase que p es un factor primo de N, por el teorema 1.1, tenemos $a^{p-1}\equiv 1 \bmod p$; esto es $p \mid a^{p-1}-1$, ahora si $(Q,p)=1$ y $p-1 \mid Q$ entonces $p \mid a^{Q}-1$ y por tanto $a^{Q}\equiv 1 \bmod p$; aquí $Q=\prod_{i=1}^{m} Q_i$ y cada $Q_i = q_i^{\beta_i}$ tal que $Q_i \leq B_1$ y $q_i Q_i > B_1$, con B_1 una cota superior no muy grande para todo $q_i^{\alpha_i}$, se tendrán $Q_1, \ldots, Q_m$ con $m \leq k$.

Ahora el calcular p se efectua como sigue, tómese un a, tal que $(a,N)=1$, hacemos $R_0=a$ y se calcula

$$R_{i+1} = R_i^{Q_{i+1}} \bmod N, \quad i=0, \ldots, m-1.$$

Al final obtenemos R_m y calculamos $A=(R_m-1,N)$; en la práctica por lo general p es A, si no, A es un número compuesto que divide exactamente a N, así que, se aplica nuevamente el método de Pollard, pero ahora A es N.

Volviendo al hecho de que $a^{Q}\equiv 1 \bmod p$, puesto que $p \mid a^{Q}-1$ y $p \mid N$ entonces $p \mid (a^{Q}-1,N)$ (aquí a^{Q} son los diferentes R_i); así es como $A=(a^{Q}-1,N)$ es p o un multiplo de él.

En caso de que A=1, se procede a ejecutar el segundo paso de que consiste el método de Pollard, en él es posible determinar un factor primo p de N, si es que p-1 es el producto de pequeños primos y un primo de mayor tamaño, es decir, si

$$p-1= s\prod_{i=1}^{k} q_i^{\alpha_i} \qquad \text{con s un número primo.}$$

Con argumentos similares a los anteriores, puede verse que en este caso $p \mid (a^{e}-1, N)$.

Es necesario contar con la siguiente información para poder aplicar el segundo paso, en el primer paso se tiene una cota B_1, aquí se hace uso de una segunda cota superior B_2 tal que $B_1 < s \leq B_2$; ahora bien se presenta un problema, si N es un entero muy grande entonces B_2 debe serlo también, esto implica que se deben conocer los primos existentes en el intervalo $(B_1, B_2]$; sean estos s_j, $j=1, \ldots, k$; teniendo estos primos calcúlese $2d_j = s_{j+1} - s_j$, así el cálculo de p se realiza como sigue :

Inicialmente $b_1 = R_m^{e} \bmod N$, con R_m el último módulo calculado en el paso 1, después

$$b_{j+1} \equiv R_m^{2 \cdot d_j} \cdot b_j \bmod N.$$

Una vez obtenidos los b_j, se calcula

$$A_t = (\prod_{i=0}^{c} (b_{t+i} - 1), N); \quad t = 1, c+1, 2c+1, \ldots, [B_2/c] \cdot c+1.$$

con c una constante entera mayor o igual a 1, que indica el número de multiplicandos a entrar en cada cálculo.

El último valor A_t es ya sea, un factor primo p de N; o si $A_t = N$ y $A_{t-1} = 1$ entonces hay que volver a intentar factorizar N en un nuevo intervalo $(B_1, B_2]$.

El método fué programado en lenguaje LISP y como prueba a dicha implementación se consideraron entre otros, a los siguientes números enteros

i) $N = 2^{99} - 1$
   Paso 1 : a=2, $B_1 = 10$, obteniendo A=7, ahora N=N/7. Después de varios intentos en este paso, no fué posible determinar otro factor primo de N, procediendo a

Paso 2 : El último residuo calculado fué $R_m=4$, así c=3, $B_1=2$, $B_2=100$, obteniendo $A_1=2147483647$ y por tanto

$N=7\cdot 2147483647\cdot 658812288653553079.$

ii) $N=2^{128}-1$

Paso 1 : Se dió a=1, $B_1=10$, obteniendo A=127 y N=N/127.

Paso 2 : $R_m=4$, c=3, $B_1=2$, $B_2=100$ así $A_1=524287$. y por tanto

$N=127\cdot 524287\cdot 1635372208527253988514343257209 59.$

iii) $N=2^{132}-1$

Se logró factorizarlo haciendo uso solamente del paso 1, obteniendo

$N=3\cdot 3\cdot 5\cdot 7\cdot 13\cdot 23\cdot 67\cdot 89\cdot 397\cdot 683\cdot 2113\cdot 20857\cdot 312709\cdot 599479\cdot 4327489.$

El siguiente número N, no fué posible factorizarlo por completo haciendo uso de los dos pasos, se tendría que emplear un intervalo $(B_1, B_2]$ más grande o intentarlo con algún otro método.

iv) $N=2^{220}-1$

$N=3\cdot 5\cdot 5\cdot 11\cdot 11\cdot 23\cdot 31\cdot 41\cdot 89\cdot 397\cdot 683\cdot 881\cdot 2113\cdot 2971\cdot 3191\cdot 201961\cdot$
$\cdot 7384233730935365278460745740239 1.$

el último término es lo no factorizado.

El método de Pollard aproximadamente requiere de $O(N^{(1/2)\alpha+\delta})$ operaciones para su próposito, siendo $0 < \alpha < 1$ y $\delta > 0$.

d.2) METODO DE MONTE-CARLO PARA FACTORIZACION.

Este es el segundo método de Pollard públicado en *1975*, la idea

para determinar un factor primo p de **N** es la siguiente :

i) Construir una sucesión de enteros (x_i), los cuáles tengan períodos módulo p.
ii) Determinar i, j tales que $x_i \equiv x_j \bmod p$.
iii) Por último determinar p.

Respecto al primer inciso, se debe seleccionar una sucesión de números (x_i) de tipo recursivo, en partícular, en este método se selecciona una función polinomial de la forma x^2+c, con c una constante diferente a 0 y a -2; la sucesión entera (x_i) se construye a partir de :

$$x_i \equiv x_{i+1} + c \bmod N, \quad i=0,1,\ldots$$

Una vez construída, se pasa a la identificación del período, si éste es corto, es relativamente fácil identificarlo; el problema se presenta cuando se trata de un período largo, la identificación del período puede hacerse confirmando la relación $x_{2i} \equiv x_i \bmod N$.

Por último el determinar p factor primo de **N**, es simplemente calcular $(N, x_{2i} - x_i \equiv \bmod N)$ incrementando i hasta obtener un valor diferente a uno.

Por ejemplo, sea **N**=91643, empleemos $x_{i+1} = x_i^2 - 1$ y $x_0 = 3$, así y para calcular el período utilizemos $y_i = x_{2i} - x_i$.

Inicialmente

$x_1 = 8$, $x_2 = 63$, $y_1 = 63-8 = 55$ por tanto $(N, y_1) = 1$.

Utilizemos las congruencias módulo **N** para no escribir grandes números, así tenemos

$x_3 = 3968$, $x_4 = 74070$, $y_2 = x_4 - x_2 = 74007$ y $(N, 74007) = 1$

$x_5 = 65061$, $x_6 = 35193$, $y_3 = x_6 - x_3 = 31225$ y $(N, 31225) = 1$

$x_7 \equiv 83746$, $x_8 \equiv 45368$, y $\equiv x_8 - x_4 \equiv 62941$ y $(N, 62941) = 113$

Por tanto N=113·811.

e) METODO "p+1".

Este método fué ideado por H. C. Williams en *1982*, sólo se darán algunos comentarios sobre el, para más detalles puede consultarse [26].

Es análogo al método **p-1** salvo que aqui, **p+1** es el producto de pequeños números primos, esto es :

$$p+1 = \prod_{i=1}^{k} q_i^{\alpha_i}$$

El algoritmo hace uso de las funciones de Lucas, que se definen de la siguiente forma Sean A, B dos enteros y λ, β las raíces del polinomio x^2-Ax+B, las funciones de Lucas estan dadas como

$$U_n(A,B) = (\lambda^n - \beta^n)/(\lambda - \beta)$$

$$V_n(A,B) = (\lambda^n + \beta^n)$$

Al igual que el método **p-1**, consta de dos pasos, en ambos el método **p+1** es más lento que **p-1**, en el primero dos veces más lento y en el segundo cuatro veces más lento; no obstante, con él, se han encontrado factores primos de números **N** que con el método **p-1** o con cualquier otro método no había sido posible determinarlos.

2. Criptografía

En el presente capítulo se resalta en primer lugar, la importancia que ha adquirido la Criptografía en la transmisión de información, así como un breve panorama de la situación que en materia de telecomunicaciones se tiene actualmente en México. Más adelante se describen algunas de las primeras técnicas utilizadas en la protección de información, no sin antes formalizar lo que se entiende por Criptografía, Encriptamiento y Desencriptamiento de mensajes.

Por último, se presenta la nueva dirección que ha tomado la Criptografía en los últimos diez años, ejemplificandolo con dos de los esquemas iniciales en esta dirección, en particular, el último esquema presentado es el objetivo de este trabajo.

2.1 Importancia de la Criptografía.

Hasta hace diez años, el problema de factorización de números enteros era de competencia exclusiva de especialistas; no se tenía aplicación alguna de interés mundial sobre este problema hasta 1978, año en que R. L. Rivest, A. Shamir y L. Adleman (RSA) del MIT, publicaron su sistema criptográfico, cuya seguridad esta basada en la enorme dificultad que se presenta al querer factorizar grandes números (veáse tabla 1.1).

Pero, ¿ cómo se ha llegado a una aplicación inmediata y de gran importancia en la teoría de números ?. Antes de dar contestación a esta pregunta, considérese los siguientes aspectos de la información.

La información puede valorarse de muy diversas maneras, por ejemplo: podría tener un costo financiero, político, militar y/o comercial si se perdiera o fuera modificada para influir en la toma de decisiones y con ello alcanzar ciertos objetivos para la cual

no fue creada, esto puede suceder cuando la información no esta físicamente asegurada, es decir, cuando se le transmite por medios de comunicación de acceso compartido que se les considera como inseguros, entre los más utilizados estan : líneas telefónicas (que pueden ser intervenidas), la transmisión por radio (microondas, pueden ser interferidas), etc.. En particular, México, con la puesta en marcha del Sistema de Satélites Morelos (SSM), en vez de reforzar la soberanía e independencia nacional, ha creado una dependencia más fuerte hacia los consorcios trasnacionales en la rama de telecomunicaciones. El porqué de lo anterior, basta señalar lo siguiente : la casi totalidad de las señales de telegrafía, telefonía, transmisión de datos, telex, teleinformática, televisión, radio, etc., que emiten tanto el gobierno como la iniciativa privada, es a través del SSM, por mencionar los más importantes : PEMEX, CFE, Secretaría de la Defensa Nacional, Secretaría de Comunicaciones y Transportes, Teléfonos de México, Telégrafos Nacionales, Banco de México, Meteorológico Nacional, la Banca Nacional, etc.; y recientemente las principales casas de Bolsa Mexicanas han iniciado la transmisión de datos por este medio.

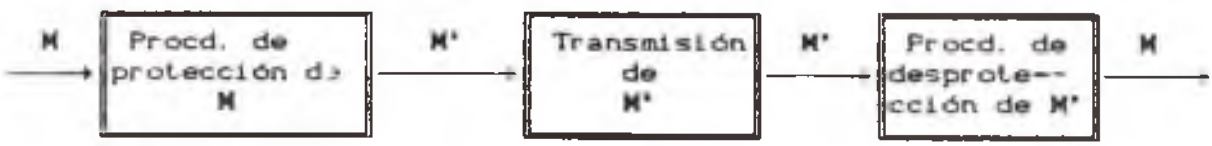
Rodolfo Neri V. [8] declaró : "es imposible evitar, que toda la información que circule por el Morelos I y II, sea conocida por otros países, especialmente los Estados Unidos"; país donde reside la compañía Hughes Aircraft Company, que construyó los satélites y parte de las estaciones terrestres; dicha compañía es el séptimo proveedor militar más grande del Pentágono y recientemente fué adquirida por la empresa más fuerte de los Estados Unios, la General Motors.

Ante esto, uno se pregunta : ¿ qué nuevo margen de independencia alcanzará el Sistema Político Mexicano, cuando todos los procedimientos de los principales aparatos del gobierno, al transmitirse por el SSM, instantáneamente sean captados por Estados Unidos ?. ¿ Qué nuevo espacio de autonomía obtendrá nuestro Estado-Nación, cuando al cruzar por el SSM datos estratégicos como son las cuotas de flujo de electricidad, el control de conducción de gas, la coordinación de los centros de aprovisionamiento de PEMEX, la exploración y explotación de las nuevas plataformas petroleras, las

condiciones meteorológicas de la agricultura regional, los montos de transacciones bancarias, la concertación de recursos fiscales, etc., puedan ser automáticamente conocidos en los países altamente industrializados ?. ¿ Qué nuevo límite de soberanía conquistaremos cuando toda la corriente de información que envían los comandos de la Armada de México y de la Secretaría de la Defensa Nacional a sus *13* zonas y *18* sectores navales respectivamente, a la infantería de marina, a los buques y aviones de la armada, al ejército,a la fuerza aérea, etc., al cruzar por el SSM, serán susceptibles de ser registrados paralelamente por la Agencia de Seguridad Nacional del Departamento de Defensa de los Estados Unidos ? [8].

Por lo anterior, el valor de la información puede llegar a ser muy grande.

Es posible que aún por canales considerados como inseguros, la información que sea transmitida resulte ser incomprensible (y por tanto, no poder obtener provecho alguno de ella) para una persona que no tenga derecho a recibirla, esto es, darle un tratamiento de protección, para que únicamente la persona autorizada pueda recobrar la información original, después de haberla protegido y transmitido por tal canal. Lo anterior puede visualizarse como :



M = mensaje.

Existen métodos para lograr lo anterior, todos ellos se enmarcan en lo que se ha llamado CRIPTOGRAFIA; una definición muy conveniente es la siguiente :

CRIPTOGRAFIA, es el estudio de sistemas matemáticos para resolver dos clases de problemas de seguridad . **privacidad y autentificación.**

Un sistema de privacidad debe prever la extracción de información por personas no autorizadas en mensajes trasmitidos por canales considerados como inseguros. Evitándolo se asegura una exitosa comunicación entre el transmisor y el receptor.

Un sistema de autentificación debe prever la transmisión no autorizada de mensajes por medio de canales públicos y que lleguen a personas autorizadas que puedan considerarlo como válido.

Por todas estas consideraciones, la criptografía ha sido de gran importancia en las comunicaciones militares, diplomáticas y recientemente en el ámbito comercial.

2.2. Tecnicas de encriptamiento y desencriptamiento de mensajes.

Un mensaje **M** (mensaje original, es decir, comprensible) será una sucesión de símbolos pertenecientes a algún alfabeto $\mathcal{U}$ (letras y/o números). Se tendrá :

$$M = (a_1 a_2 \ldots a_k \mid a_i \in \mathcal{U},\ \forall\ i)$$

Una **transformación criptografica,** opera sobre cada símbolo, produciendo un símbolo clave, éste cae en un alfabeto $\mathcal{B}$ (generalmente $\mathcal{U} = \mathcal{B}$) El resultado de aplicar la transformación criptografica a un mensaje **M**, es un mensaje C que resulta ser incomprensible; se tiene :

$$T : \mathcal{U} \longrightarrow \mathcal{B} \qquad y$$

$$C=T(M)=(T(a_1), T(a_2), \ldots, T(a_k) \mid T(a_i) \in \mathcal{B},\ \forall\ i)$$

Se entenderá por **ENCRIPTAMIENTO (E)**, a la acción de aplicar una o más transformaciones criptográficas a un mensaje M, con el fin de producir un mensaje en clave C y por **DESENCRIPTAMIENTO (D)**, a la acción de aplicar una o más transformaciones criptográficas a C para recuperar **M**.

Los **ESPACIOS DE ENCRIPTAMIENTO (𝔈)** y **DESENCRIPTAMIENTO (𝔇)**, serán conjuntos de transformaciones criptográficas.

Una **LLAVE (L)**, será un valor específico que dirá que transformación criptográfica debe aplicarse. Un **ESPACIO DE LLAVES 𝔏**, será el conjunto de valores que puede tomar L.

Se define a un **SISTEMA CRIPTOGRAFICO 𝔗**, como una familia de transformaciones criptográficas :

$$\mathfrak{T} = \{T_L : L \in \mathfrak{L}\}$$

En términos generales, un sistema criptográfico (véase figura 2.1) opera sobre un usuario, que desea transmitir un mensaje **M** a un receptor por medio de un canal de comunicación considerado como inseguro, en este caso, una tercera persona puede tener acceso a dicho canal y por tanto interceptar a **M**, con el consecuente riesgo de que exista la posibilidad de conocerlo y/o modificarlo a su propia conveniencia; además, también es posible que pueda emitir un mensaje que sea considerado como válido y con ello alcanzar ciertos logros que le beneficien.

Puede resumirse en tres puntos, los principales objetivos de una persona no autorizada para transmitir y/o recibir mensajes.

a) Conocer el mensaje **M**.
b) Alterar **M** por algún otro mensaje **M'** y que éste sea aceptado por el receptor, suponiendo que proviene del transmisor.

c) Iniciar comunicación con el receptor, suplantando al transmisor autorizado.

SISTEMA CRIPTOGRAFICO

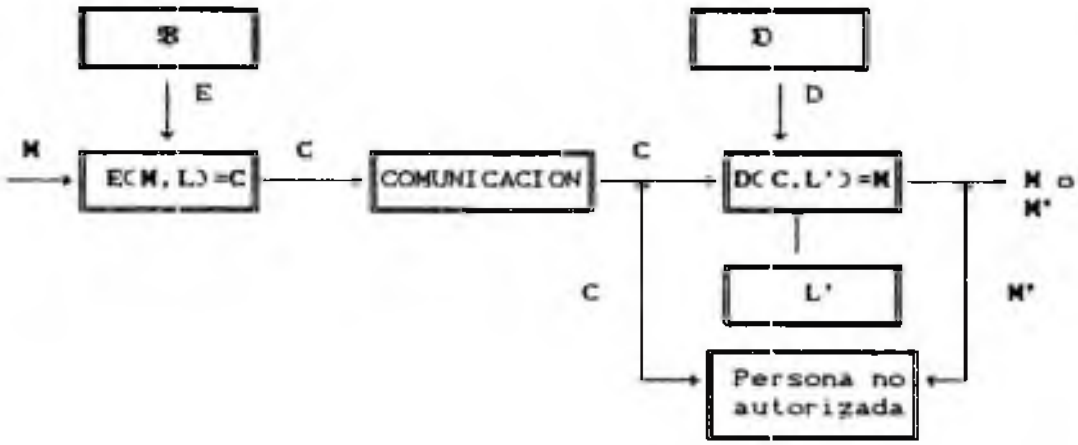


figura 2.1

Ahora bien, la mayoria de las técnicas de encriptamiento basan su seguridad (esto es, que una persona no autorizada, no sea capaz de romper el sistema) en la mínima cantidad de tiempo de cómputo que deba invertir una persona, para poder desencriptar a C sin conocer la llave L; tal estimación del tiempo resulta ser un problema de la teoría de complejidad y análisis de algoritmos.

TECNICAS.

a) El más sencillo de todos los sistemas criptográficos, es la sustitución mono-alfabética; consiste en que cada letra de un mensaje M es reemplazada por otra, esto puede llevarse a cabo por medio de la tabla VIGENERE :

Aqui $\mathfrak{L}$ = (0,1, . . ., 25) y $\mathfrak{U}$ = (A,B, ,Z)

La tabla de Vigenere :

llave	A	B	C	D	E	F	G	.	.	.		Z
0	A	B	C	D	E	F	G	.	.	.		Z
1	B	C	D	E	F	G	H	.	.	.	Z	A
2	C	D	E	F	G	H	I	.	.	Z	A	B
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
25	Z	A	B	C	D	E	F	.	.			Y

La transformación criptografica se define como :

$$T_i : \mathcal{U} \longrightarrow \mathcal{U} \qquad 0 \leq i \leq 25.$$

$$T_i(x) = y , \qquad x, y \in \mathcal{U}$$

Por ejemplo M = QUETZAL

T_2(QUETZAL) = SWGVBCN.

Este sistema es fácil de romper.

b) El encriptamiento por cambio de posición, consiste de una permutación P de n símbolos (P es la llave) y cada bloque sucesivo de n símbolos en el mensaje son reordenados haciendo uso de P.

Considérese a :

M = SI YO NUNCA DESAPARECIERA

y

$$P = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 7 & 9 & 5 & 2 & 6 & 1 & 4 & 3 & 8 & 11 & 10 \end{pmatrix}$$

La permutación P indica que el primer carácter de M pasa a la séptima posición, el segundo carácter a la novena posición, etc., por tanto:

$E_P(M)$ = UOCNYNSAIEDEAICPRSEAAR.

A simple vista podría considerarse muy simple esta técnica; pero nada impide tomar como llave a una serie de permutaciones $P_1, \ldots, P_k$ y aplicarlas sucesivamente a M, de tal forma que el mensaje encriptado C adquiera una forma muy "oscura" de descifrar, no obstante existen algoritmos computacionalmente factibles para romper el sistema.

c) En el año 1929, L. S. Hill mostró que los sistemas criptográficos pueden ser formalizados por medio de transformaciones lineales.

Un mensaje M es primeramente transformado en un vector de dimensión n, siendo n el número de símbolos de que consta dicho mensaje. Bajo la convención A=01, B=02, ..., Z=26, se tendrá M=CASTRO=(03,01,19,20,18,15); ahora bien, el encriptamiento y desencriptamiento se hace en base a transformaciones lineales invertibles, puesto que una transformación puede ser representada por una matríz, las operaciones sobre M se realizan por multiplicación de matrices que representan a las transformaciones seleccionadas; éstas matrices son no-singulares (ello asegura la existencia de la matríz inversa utilizada para recuperar M).

Por citar un ejemplo : Sea M=CASTRO=(03,01,19,20,18,15), puesto que se tienen 26 letras, la aritmética que se utilizará será la multiplicación de matrices módulo 26.

Sea $A=\begin{pmatrix} 2 & 15 \\ 5 & 24 \end{pmatrix}$ la matríz de encriptamiento, puede verse que $A^2=I$; por tanto ella misma es la matríz para desencriptar.

Encriptamiento de M

(C,A)=(03,01)(A)=(11,17)=(K,Q)

(S,T)=(19,20)(A)=(08,11)=(H,K)

(R,O)=(18,15)(A)=(07,06)=(G,F)

así, $E_A(M)$=KQHKGF.

Las anteriores técnicas fueron algunas de las primeras en ser aplicadas, existen muchas otras más que no serán mencionadas aquí, fijaremos la atención en las más recientes que han marcado un nuevo camino en la Criptografía.

2.3. Nuevo esquema en Criptografía.

En el año 1976, W. Diffie y M. Hellman de la Universidad de Stanford en los Estados Unidos, publicaron un nuevo esquema para sistemas criptográficos, al cual llamaron **SISTEMA CRIPTOGRAFICO DE LLAVE PUBLICA (SCLP)**, ya que en él, no es necesario intercambiar la llave, lo definieron de la siguiente forma :

Un SCLP es un par de familias $\langle E_L \rangle$ y $\langle D_L \rangle$ de algoritmos que representan las transformaciones invertibles :

$$E_L : \langle M \rangle \longrightarrow \langle M \rangle$$

$$D_L : \langle M \rangle \longrightarrow \langle M \rangle$$

donde $\langle M \rangle$ es un espacio finito de mensajes, tal que se cumplen las siguientes condiciones :

i) Si $C=E_L(M)$ entonces $M=D_L(C)$.
ii) E_L y D_L son fáciles de computar.
iii) El hacer público a E_L no compromete a D_L, esto es, la derivación de D_L a partir de E_L es computacionalmente infactible.
iv) Para todo $L \in \mathfrak{L}$, es factible computar E_L y D_L a partir de L.

Es por la tercera propiedad, por lo que la llave que utiliza el transmisor para encriptar puede hacerla pública sin que con ello comprometa la seguridad de la llave para desencriptar el mensaje. La implementación de estos sistemas criptográficos se hace por medio de

problemas difíciles de computar, tal y como son las funciones de "un solo sentido".

Una función f se dice ser de **UN SOLO SENTIDO**, si existe su inversa y f es fácil de computar, pero computacionalmente no es factible calcular f^{-1}.

Una función f es llamada una función **TRAMPA** en un solo sentido, si f^{-1} es fácil de computar una vez que cierta información privada es conocida y sin tener conocimiento de que f es de un solo sentido.

Nótese que una función f y su inversa f^{-1} definidas como arriba, pueden jugar el papel de E y D en el esquema SCLP.

Es bajo este esquema, que los sistemas propuestos están basados en problemas matemáticos de alta complejidad computacional y que cumplen con los anteriores requerimientos; se mencionarán solamente los dos más importantes.

a) ESQUEMA DE MERKLE-HELLMAN.

El presente esquema está basado en el problema combinatorio conocido como "problema de la suma de un subconjunto"; consiste en tener un vector a de dimensión n, tal que $a_i \in \mathbb{N}\ \forall i$ y a S como la suma de ciertos elementos de a; de esta forma el problema es el determinar qué subconjunto de a da como suma a S, esto es, determinar el vector x con $x_i = 0,1\ \ \forall i$, tal que

$$S = a \cdot x = \sum_{i=1}^{n} a_i x_i \qquad (2.1)$$

Sirva de ejemplo lo siguiente :

Sean a=(13,1030,1305,5,1964,1962) y S=3944. Para determinar una solución x de (2.1), no existe algún método obvio. En nuestro ejemplo, después de algunas combinaciones con los elementos de a, se puede obtener la solución x=(1,0,0,1,1,1). Cabe señalar que siempre se tendrán 2^n subconjuntos de a y esto, cuando n es grande, hace que resulte infactible el ir probando cada subconjunto; por otra parte, x no necesariamente es unico, incluso puede o no existir, en el ejemplo S=1817, no tiene solución con a.

R. Karp ha demostrado que el problema de determinar x, es de complejidad exponencial, es aquí donde radica la seguridad del esquema. Para que resulte infactible computar x, Merkle y Hellman sugieren tomar n≥100.

Los mensajes M bajo este esquema, son vectores binarios $x=M=(b_1, \ldots, b_n)$ con $b_i=0,1\ \forall i$ y la llave de encriptamiento es el vector $a=L=(a_1, \ldots, a_n)$ con $a_i \in \mathbb{N}\ \forall i$. El algoritmo de encriptamiento para M es :

$$S=E_L(M)=\sum_{i=1}^{n} a_i b_i$$

El vector a es calculado de la siguiente forma :

Selecciónese dos enteros p y q que cumplan

i) p y q deben ser grandes, con p<q.
ii) p debe ser invertible módulo q.

Teniendo p y q, tómese un vector a' tal que $a'_i \geq \sum_{j=1}^{i-1} a'_j$, además se debe tener que $a'_n < q$, a continuación calcule los elementos a_i de a bajo la relación :

$$a_i = p \cdot a'_i \bmod q \ .$$

los números p y q deben ser guardados en secreto.

Para desencriptar, se hace uso del siguiente algoritmo :

Calcule $S' = p^{-1} \cdot S \bmod q$, luego

a) Si $S'_n \geq a'_n$ entonces $b_n = 1$.

en otro caso $b_n = 0$.

b) Para j=n-1 hasta 1

si $S' - \sum_{i=j+1}^{n} b_i \cdot a'_i \geq a'_j$ entonces $b_j = 1$.

en otro caso $b_j = 0$.

Ejemplo :

Sean n=6, $a' = (43, 127, 235, 721, 1333, 2737)$, q=6763 y p=12541, por tanto p^{-1}=666 ya que $p \cdot p^{-1} \equiv 1 \bmod q$, nótese que $a'_i \geq \sum_{j=1}^{i-1} a'_j$

Se contruye a bajo la relación :

$a_i \equiv 12541 \cdot a'_i \bmod 6773$, obteniendose

$a = (4986, 3402, 5230, 6693, 5780, 2492)$.

este vector se hace público y es utilizado para el encriptamiento de mensajes.

Sea S=3402+6693+2492=12587 el encriptamiento de un mensaje.

Para desencriptar, calculemos

$$S' = p^{-1} \cdot S \bmod q$$
$$= 666 \cdot 12587 \bmod 6763$$
$$= 3585.$$

Puesto que $S'=3585>2737=a'_6$, entonces $b_6=1$

Para j=5 hasta 1

$$\text{si} \quad S' - \sum_{i=j+1}^{6} b_i \cdot a'_i \geq a'_j \quad \text{entonces} \quad b_j=1.$$

$$\text{en otro caso } b_j=0.$$

j=5; $3585-(1\cdot 2737) = 848 < 1333 \rightarrow b_5=0.$

j=4; $3585-(0\cdot 1333+2737) = 848 > 721 \rightarrow b_4=1.$

j=3; $3585-(1\cdot 721+2737) = 127 < 235 \rightarrow b_3=0.$

j=2; $3585-(0\cdot 235+3458) = 127 \geq 127 \rightarrow b_2=1.$

j=1; $3585-(127+3458) = 0 < 43 \rightarrow b_1=0.$

por tanto x=(0,1,0,1,0,1), que es solución a S=a·x.

b) ESQUEMA DE RIVEST-SHAMIR-ADLEMAN (RSA).

El presente esquema da respuesta a la pregunta inicial del capítulo. En el capítulo anterior se vió la infactibilidad de factorizar números de más de 200 digitos, si se utilizara el algoritmo de Schoroppel, se tendría que esperar varios miles de millones de años para ello; la seguridad del esquema RSA está basada es esta dificultad.

Los mensajes toman forma de números en el rango [0,n-1]; donde n es el producto de dos números primos p y q de no menos de 100 digitos cada uno, de esta forma se asegura que n será de a lo menos 200 dígitos. El número n se hace público.

Sea r=(p-1)·(q-1), debe determinarse un entero d mayor al máximo de p y q, tal que (d,r)=1, en seguida calcúlese el entero e que cumpla e·d=1 mod r, esto es, e es el inverso multiplicativo de d módulo r.

El mensaje M deberá ser partido en bloques, cada uno de ellos será un entero entre 0 y n-1. El mensaje encriptado C a partir de M, se calcula de la forma siguiente :

$$C=E_e(M)\equiv M^e \bmod n \qquad (2.2)$$

Para recuperar M a partir de C :

$$M=D_d(C)\equiv C^d \bmod n \qquad (2.3)$$

Puesto que se hace uso de exponenciaciones, es conveniente realizarlas mediante el siguiente algoritmo, conocido como "exponenciación por cuadrados repetidos y multiplicación" .

a) Calcúlese la representación binaria de e; sea esta $e_k\ e_{k-1}\ \ldots\ e_1 e_0$.
b) C=1.
c) Para i=k hasta 0
   $C=C^2 \bmod n$.
   Si e_i=1 entonces $C=C\cdot M \bmod n$.

Como ejemplo considérese lo siguiente :

Sean p=661 y q=691, así n=p·q=456751, por tanto r=455400; sea d=709 con el se tiene (d,r)=1 y por tanto e=63589.

Con n=456751, seis dígitos, el mensaje M puede ser partido en bloques de tres carácteres , tomando la convención : blanco=10, A=11, B=12, . .. Z=36.

Encriptemos el mensaje M=AQUI NADIE VIVIRA PARA SIEMPRE; partido queda como 112731 191024 111419 151032 192219 281110 261128 111029 191523 262815. La representación binaria de e=63589=1111100001100101, así empleando el algoritmo anteriormente descrito para encriptamiento de mensajes, M resulta .

M_1 =112731 ⟶ C_1 =380598.
M_2 =191024 ⟶ C_2 =133250.
M_3 =111419 ⟶ C_3 =456675.
M_4 =151032 ⟶ C_4 =257835.
M_5 =192219 ⟶ C_5 =26730.
M_6 =281110 ⟶ C_6 =151001.
M_7 =261128 ⟶ C_7 =438504.
M_8 =111029 ⟶ C_8 =291568.
M_9 =191523 ⟶ C_9 =203190.
M_{10} =262815 ⟶ C_{10} =332129.

Por tanto C=380598 133250 456675 257835 26730 151001 438504 291568 203190 332129.

Para el desencriptamiento lo único que cambia entre (2.2) y (2.3) es M por C y d por e; de esta forma es posible utilizar el mismo algoritmo de encriptamiento para desecriptar C, empleando ahora d en lugar de e.

3. Aplicaciones

Se presenta una implementación del sistema Criptográfico RSA, así como dos algoritmos para generar grandes números con una fuerte certeza de ser números primos, además la implementación del algoritmo "p-1" para la factorización de números enteros. Todas las implementaciones fueron hechas en el lenguaje LISP, en cada algoritmo se da su orden de complejidad.

3.1. Algoritmos para la generacion de numeros primos.

Tanto el estudio como la implementación del algoritmo ideado por Cohen y Lenstra es muy complejo y dado que el objetivo del presente trabajo es mostrar la utilidad de los números primos a la Criptografía, se ha hecho uso de dos algoritmos probabilisticos para generar grandes números con una fuerte certeza de sér primos, esto es, cuando los algortimos determinan que un número **N** es compuesto, la aseveración es siempre cierta, pero cuando determinan que el número es primo, se hace con una cierta probabilidad de error, que aunque es menor a 1/2 siempre existira la posibilidad de trabajar con un número compuesto como número primo.

El primer algoritmo que se presenta, fué ideado por **R. SOLOVAY** y **V. STRASSEN** en 1977, en él, el número **N** a ser probado no tiene forma especial alguna, he aquí el algoritmo descrito en un seudo-código.

Sea **N** un entero impar mayor que la unidad.
Sean $a_1, \ldots, a_k$ números enteros, escogidos aleatoriamente en el intervalo [1..**N**-1].
Para i=1 hasta k haga
inicio
  Si (a_i, N)=1 entonces
    inicio
      $\delta = a_i^{(N-1)/2} \bmod N$;
      $\gamma = \left(\frac{a_i}{N}\right)$;

```
                    Si δ=ψ entonces número="posible_primo";
                        en otro caso
                            número="compuesto"; termina;
                fin
                en otro caso número="compuesto"; termina;
        fin
        imprime(número).
```

Aquí $\left(\frac{\alpha_i}{N}\right)$ representa el símbolo de **JACOBI**.

Calcular el máximo común divisor de α_i y **N**, puede llevarse ha cabo con el algoritmo de Euclides, lo cuál requiere aproximadamente de $1.5 \log_2 N$ operaciones [22], con ello se determina si α_i y **N** son primos relativos. Por otra parte, el cálculo de δ puede hacerse por el algoritmo potencia que ha continuación se describe.

```
Sea (b_0 ... b_r) la representación binaria de N-1/2.
K=1
Para i=r hasta 0 haga
inicio
   K=K^2 mod N;
   Si b_i=1 entonces K=K·α_i mod N;
termina.
```

Este algoritmo hara uso de a lo más $2\log_2 N$ multiplicaciones y $2\log_2 N$ divisiones [22].

Antes de exponer el algoritmo para el cálculo del símbolo de Jacobi, se hace necesario dar algunas definiciones y resultados que sobre de él se tienen.

En la teoría de números se estudian varios tipos de congruencias, en partícular las congruencias cuadráticas, es decir, congruencias de la forma $x^2 \equiv a \bmod m$ siendo $(a,m)=1$; si esta congruencia admite

solución, a a se le llama **residuo cuadrático**, en caso contrario **no-residuo cuadrático**.

El símbolo de **LEGENDRE**, denotado por $\left(\frac{a}{p}\right)$ con $(a,p)=1$, se define como :

$$\left(\frac{a}{p}\right) = \begin{cases} 1; \text{ si } a \text{ es un residuo cuadrático} \\ -1; \text{ en otro caso.} \end{cases}$$

Sea P un número impar y sea $P=p_1 \cdots p_r$ su descomposición canónica, supongase también que $(a,P)=1$, el símbolo de Jacobi se define :

$$\left(\frac{a}{P}\right) = \left(\frac{a}{p_1}\right) \cdots \left(\frac{a}{p_r}\right)$$

con $\left(\frac{a}{p}\right)$ símbolos de Legendre; así el símbolo de Jacobi toma un valor $(1,-1)$.

El porque del funcionamiento del algoritmo de Solovay y Strassen lo da el siguiente criterio, debido a Euler.

Teorema 3.1.

Si $(a,p)=1$ y si p es un número primo impar, entonces

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \bmod p \quad \forall a \in \mathbb{Z}.$$

Puesto que $\left(\frac{a}{p}\right) = (1,-1)$, entonces el criterio anterior indica que a es un residuo o un no-residuo cuadrático módulo p, además si p es un número primo, entonces los símbolos de Legendre y Jacobi coinciden.

El símbolo de Jacobi puede ser calculado haciendo uso de la ley reciproca.

Teorema 3.2.

Si **m** y **n** son enteros positivos impares y si $(a,m)=1$ y $(b,n)=1$ entonces

$$(i)\quad \left(\frac{a}{m}\right)\left(\frac{b}{m}\right)=\left(\frac{ab}{m}\right)$$

$$(ii)\quad \left(\frac{-1}{n}\right)=(-1)^{(n-1)/2}$$

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

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

Se presentan dos algoritmos para el cálculo del símbolo de Jacobi, el primero de manera recursiva que resulta ser muy elegante, pero que en cuanto a_i como **N** son muy grandes resulta inadecuado su uso, el otro es en forma iterativa, en ambos van como entrada a_i y **N**, el valor del símbolo de Jacobi se deposita en la variable "sj".

Forma recursiva

```
Jacobi(a_i ,N)
inicio
  Si a_i =1 entonces sj=1
            en otro caso
            inicio
            si a_i es par entonces sj=Jacobi(a_i /2,N)*(-1)^((N^2-1)/8)
                          en otro caso
```

```
                    sj=Jacobi(N mod a_i ,a_i )*(-1)^((a_i-1)*(N-1)/4)

            fin
    fin.
```

Forma iterativa

```
    Jacobi(a_i ,N)
    inicio
      u=0; i1=0; u1=0; x1=0; x2=0; z=0; m=0; a1=a_i
      nn=N mod 8; n1=N;
      mientras n1 > 1 haga Sjacobi_aux(a_i , N
    fin
    Sjacobi_aux(a_i ,N)
    inicio
      i1=0;
      mientras a1=u*2 haga
      inicio
         u=a1 div 2;
         i1=i1+1;
         a1=u;
      fin
      si i1 es impar entonces m=m(nn^2-1) div 8;
      u1=a1 mod 4; x1=nn-1; x2=u1-1;
      m=(x1*x2) div 4 + m; z=n1 mod a1; n1=a1; a1=z;
      nn=n1 mod 8;
      si n1 > 1 entonces regresa;
      m=m mod 2;
      si m = 0 entonces sj=1;
               en otro caso sj=-1;
      regresa
    fin.
```

El número de operaciones involucradas para el cálculo del símbolo de Jacobi, es aproximadamente igual al del algoritmo de Euclides, es decir, $1.5 \log_2 N$; así el numero total de operaciones en el algoritmo

de Solovay y Strassen no es mayor a $7k \log_2 N$, con k el número de a_i seleccionados en [1, N-1]; se recomienda que k=100.

En su artículo, demuestran que el conjunto

$$\mathcal{G}=\{a+N\mathbb{Z} \mid a\in\mathbb{Z} \text{ y } (a,N)=1,\ a^{(N-1)/2} \equiv \left(\frac{a}{m}\right) \bmod N\}$$

es un subgrupo del grupo de unidades Z_N^* de Z_N, y $|\mathcal{G}| \leq 1/2\ |Z_N^*| \leq (N-1)/2$, esto es, a lo más la mitad de los números en [1, N-1] nos llevarán a la aseveración de que N es primo.

El segundo algoritmo que se presenta, fué ideado por **MICHAEL O. RABIN**. Antes de presentar el algoritmo en seudo-código, se dan algunas definiciones y resultados que muestran la efectividad de dicho algoritmo.

Definición 3.1. Sea N un entero, se define la siguiente condición denotandola por $\mathcal{W}_N(b)$ para b un número entero

i) $1 \leq b \leq N$.

ii) a) $b^{N-1} \not\equiv 1 \bmod N$, ó

b) $\exists i$ tal que $2^i | (N-1)$ y $1 < (b^{(N-1)/2^i}-1, N) < N$.

A tal número se le llama un "**testigo**" de la no primicidad de N, esto porque si $\mathcal{W}_N(b)$ se cumple para algún b, entonces N es un número compuesto ya que si la condición ii) a) se cumple, esto hara que la condición del pequeño teorema de Fermat sea violada, así N no es un número primo; si la condición ii) b) se cumple, implica que N tiene un divisor y por tanto es un número compuesto. En su artículo Rabin demuestra el siguiente teorema.

Teorema 3.3.

Si N>4 es un número compuesto, entonces

$$\frac{3(N-1)}{4} \leq C(\{b \mid \mathcal{W}_N(b) \text{ es cierta}\}).$$

donde C(A) denota el número de elementos que forman a A.

Por la primera condición del teorema 3.2 y este último teorema, se tiene que no más de un cuarto de números $1 \leq b \leq N$ son no testigos, esto es, hay muchos números "testigos" de que N sea un número compuesto. Es por esto último que si se toman $b_1, \ldots, b_k$ números aleatorios en [1,N-1] y si para cada uno del ellos la definición no se cumple, entonces hay una fuerte certeza de que N sea un número primo.

Para el algoritmo de Rabin, se pide que $N-1=2^l m$, con m un número impar. Sean $b_1, \ldots, b_k$ números escogidos aleatoriamente en el intervalo [1,N-1].

```
inicio
    número="primo"
    para i=1 hasta k haga
    inicio
      res=b_i^(N-1) mod N
      si res<>1 entonces
                  P_N(b_i) es cierta;
                  número="compuesto";
                  termina
                en otro caso
                  para j=1 hasta l haga
                  inicio
                    x1=b_i^(2^j m); x2=x1 mod N;
                    x3=(x2,N);
                    si x3 e (1,N) entonces
                                  P_N(b_i) es cierta;
                                  número="compuesto";
                                  termina;
                  fin
    fin
  imprime(numero);
fin.
```

A simple vista puede notarse que el presente algoritmo requiere de

más operaciones que el anterior, en su artículo, Rabin da un número de operaciones igual a $k(2\log_2 N + l \cdot \log_2 N)$ y el error de aseverar que un número N es primo (siendo éste un número compuesto) es menor a $1/2^{2k}$.

LOUIS MONIER demostró en 1980, que el algoritmo de Rabin es más eficiente que el de Solovay y Strassen.

Se ha estado trabajando de la siguiente forma, se propone un número **N**, se le somete a una serie de divisiones entre los números primos comprendidos en el rango [3,7000], evitando con ello de que **N** tenga pequeños factores primos ya que una elección de tal **N** sería objeto de una fácil factorización por el método "p-1". Si **N** pasa estas divisiones, se le aplican los dos algortimos descritos con anterioridad y con ello se tiene una fuerte certeza de que **N** es efectivamente un número primo.

Aplicando lo anterior, se han encontrado los siguientes números, ellos serán utilizados en la implementación del sistema Criptográfico RSA.

D=71104006471111046464007046464064007111104647111104640020417

P=142222092929292929422222092928000129292942220929294220935297

Q=64000006471046464646464646464646464647111110470464711040716l7

Tanto D como Q constan de 59 dígitos y P de 60 dígitos, así $N=P \cdot Q$ constará de 119 dígitos; por último el número E tal que $E \cdot D \equiv 1 \bmod (P-1)(Q-1)$ fué calculado por el algoritmo de Euclides, arrojando el valor de

E=209840750886087897958502171785106276725618732827796535968612771366888l436179248.

Los números P,Q, y D fueron sometidos al algoritmo de

factorización ideado por Pollard y conocido como algoritmo "p-1" para la factorización de números enteros; éste se basa en el hecho de que si **N** tiene un factor primo **p**, se cumple

$$p-1=\prod_{i=1}^{k} q_i^{\beta_i}$$

con q_i distintos primos y $\beta_i \in \mathbb{N}$ $\forall i$. Se asume que cada $q_i^{\beta_i} < B_1$, con B_1 una cierta constante dada.

Es necesario construir una lista que conste de todas las potencias primas Q_i tales que $Q_i \leq B_1$ y $q_i Q_i > B_1$; por ejemplo, si $B_1=15$, entonces $Q_1=8=2$, $Q_2=9=3$, $Q_3=5$, $Q_4=7$, $Q_5=11$ y $Q_6=13$. Sean **m** potencias primas Q's.

PASO 1.

Sea a un entero tal que (a,N)=1
$R_0=a$
para i=0 hasta m-1 haga $R_{i+1} \equiv R^{Q_{i+1}} \bmod N$;
p=(R_m -1,N);
imprime("un factor de N es" p)

Los argumentos que hacen válido este algoritmo, fueron expuestos en el anterior capítulo. En el paso 2 del algoritmo, se asume que el factor **p** de **N** tiene la forma

$$p-1=s\cdot\prod_{i=1}^{k} q_i^{\beta_i}$$

con q_i y β_i igual que antes y **s** un número primo que cumple $B_1 < s \leq B_2$, con B_2 una nueva cota dada. Deben de construirse dos listas, una que contenga a todos los primos s_j que cumplan $B_1 < s_j \leq B_2$ y otra de la forma $d_j = s_{j+1} - s_j$, digamos j=1,. . . ,r; obtenido lo anterior se procede a :

PASO 2.

$b_i = R_m^{*^i} \bmod N$;

para $j=1,\ldots,r$ haga $b_{j+i} = R_m^{2^j} \cdot b_j \bmod N$;

para $t=1, c+1, \ldots, [^{b}2/c]c+1$ haga

inicio

$prod = \prod_{i=0}^{c} (b_{i+i} - 1)$

$G_t = (prod, N)$

imprime ("un posible factor de N" G_t)

fin

Aqui c es una constante entera mayor a la unidad. Con este algoritmo no se encontro algún factor primo para los números P, Q y D, así que crece aún más la posibilidad de que dichos números generados por los algoritmos de Solovay, Strassen y Rabin sean efectivamente números primos.

La complejidad del algoritmo de Pollard, está dada por el siguiente teorema, enunciado y demostrado por él mismo en 1974, su demostración es muy compleja, el lector puede consultarla en [16].

Teorema 3.4.

Sean $0<\alpha<1$ y $b>0$; existe una máquina de Turing que puede determinar si dado un gran numero N, éste tiene factores primos no mayores a N^{α} y en caso afirmativo,es encontrado en un número $O(N^{(\alpha/2)+b})$ de operaciones.

Algunos ejemplos del uso de este algoritmo, han sido dados en el capítulo anterior

Obtenidos P,Q y D, es posible encriptar mensajes M por medio del sistema RSA.

3.2. Sistema Criptográfico RSA.

Hasta la fecha, este sistema para encriptamiento y desencriptamiento de mensajes a resultado ser seguro, de hecho, se ha demostrado que el problema que implica el querer romper el sistema, es equivalente al problema de factorizar el número **N**; tan difícil resulta, que aún el demostrar la existencia de un algoritmo de tiempo polinomial para la factorización de números enteros sigue siendo un problema abierto en las matemáticas.

La implementación fué diseñada para computadoras personales (PC's), por la capacidad de memoria se ha estado trabajando con números **N** de no más de 120 digitos, que aún así, su factorización requerirá de no menos de un siglo de tiempo de máquina según la tabla 1.1. Con este tipo de números, es posible manejar mensajes **M** partidos en lotes de 60 caracteres; puede tomarse cualquier convención para relacionar los caracteres que forman el alfabeto **U** (ver pag.23) y un subconjunto de números naturales; en particular se tomo la convención más simple, 10↔blanco, 11↔A,..., 36↔Z, 37↔0,..., 46↔9; así el alfabeto utilizado es **U**=⟨A,...,Z,0,...,9⟩.

A continuación se exponen algunas ideas generales para la implementación del sistema RSA.

Sea $M=a_1,\ldots,a_k$ un mensaje tal que $a_i \in U\ \forall i$.

Sean **P**, **Q**, **D** y **E** números que cumplan las condiciones pedidad por RSA; recordamos que el mensaje encriptado **C** a partir de **M** se obtiene

$$C=M^E \bmod N.$$

y la recuperación de **M** a partir de **C**

$$M=C^D \bmod N.$$

Sea **nd** el número de digitos de que consta **N**, puesto que se hace uso de dos digitos por cada carácter, se tendrán lotes de **tl**=**nd** div 2

caracteres. Dependiendo de la convención tomada para la enumeración de los caracteres del alfabeto U, se transforma $M=\alpha_1 \ldots \alpha_k = d_1 \ldots d_k$;con $d_i = \langle 11, \ldots, 46 \rangle$ que es nuestro caso.

A continuación se debe partir a M en bloques de tl caracteres, en caso de que el último bloque no conste exactamente de ellos, se debera rellenar del número que indique el símbolo blanco.

Una vez partido M en bloques, digamos $M=M_1 \ldots M_s$; $M_i = d_1 \ldots d_{tl}$ $\forall i$, es posible encriptar el mensaje, obteniendo

$$C_1 = M_1^E \bmod N$$

$$\vdots$$

$$C_s = M_s^E \bmod N.$$

así $C=C_1 \ldots C_s$. El algoritmo tanto para encriptar como para desencriptar fué dado en el anterior capítulo.

Como una de tantas pruebas a la que fue sujeto el programa, se utilizaron los números propuestos en la página 42 ; el mensaje encriptado y después desencriptado fué

YO NEZAHUALCOYOTL LO PREGUNTO
ACASO DE VERAS SE VIVE CON RAIZ EN LA TIERRA
NO PARA SIEMPRE EN LA TIERRA SOLO UN POCO AQUI
AUNQUE SEA DE JADE SE QUIEBRA
AUNQUE SEA DE ORO SE ROMPE
AUNQUE SEA PLUMAJE DE QUETZAL SE DESGARRA
NO PARA SIEMPRE EN LA TIERRA SOLO UN POCO AQUI
PERCIBO LO SECRETO LO OCULTO
OH VOSOTROS SENORES ASI SOMOS
DE CUATRO EN CUATRO NOSOTROS LOS HOMBRES
TODOS HABREMOS DE IRNOS
TODOS HABREMOS DE MORIR EN LA TIERRA
COMO UNA PINTURA NOS IREMOS BORRANDO

COMO UNA FLOR NOS IREMOS SECANDO
AQUI SOBRE LA TIERRA
COMO VESTIDURA DE PLUMAJE DE AVE ZACUAN
DE LA PRECIOSA AVE DE CUELLO DE HULE
NOS IREMOS ACABANDO
MEDITADLO SENORES AGUILAS Y TIGRES
AUNQUE FUERAIS DE JADE
AUNQUE FUERAIS DE ORO
TAMBIEN ALLA IREIS
AL LUGAR DE LOS DESCARNADOS
TENDREMOS QUE DESAPARECER
NADIE HABRA DE QUEDAR
ESTOY EMBRIAGADO LLORO ME AFLIJO PIENSO DIGO
EN MI INTERIOR LO ENCUENTRO
SI YO NUNCA MURIERA
SI NUNCA DESAPARECIERA
ALLA DONDE NO HAY MUERTE
ALLA DONDE ELLA ES CONQUISTADA
QUE ALLA VAYA YO
SI YO NUNCA MURIERA
SI YO NUNCA DESAPARECIERA
A DONDE IREMOS DONDE LA MUERTE NO EXISTE
MAS POR ESTO VIVIRE LLORANDO
QUE TU CORAZON SE ENDERECE
AQUI NADIE VIVIRA PARA SIEMPRE
AUN LOS PRINCIPES A MORIR VINIERON
HAY INCINERAMIENTO DE GENTE
QUE TU CORAZON SE ENDERECE
AQUI NADIE VIVIRA PARA SIEMPRE.

Obteniendo el mensaje encriptado (cada C esta formado por un número que consta de dos renglones).

509904657392803463302881649213194587323827490498297807082589530874871237409068525964727899689833714422678203145930421760204370434955575474040543994574827523588495506631005180350575195983919504837191353100133625395372599685421687585299555

526643886745213731 2887681 4494741 749987211 87473261 76961 6084547642430
7484733491 8309992371 8371 3442354768853819035569341 1

3763345857875262846729722017931 1 703570915455935111 81 1509077048758820
5438589584369317263230158089852510058859866712 84950

693923851 762872879386219521 9593365157700631 7423082311 2823418679722 1
2662032812932657235565111 2333478204789234658850 13761

1 3696177475433743248121 6845537747347404274302944915597772752573239 2
9326446848155386409934446987849330575135 1 7037924323

1 647927626846701 5896631 4864486589908668172780203019309762499929451 3
21 944491 9296241 0459908608437383847294637384525901 5

359569751 026971 87723590641 2621 7291 87973521 9926694228738276975847646
9579309396461 92561 61 0202951 5438998495006432994243 80

49588780221 471 532079878620184163201 71 0545266638513318776961 7885781 8
31 426847901 4009888862325968096033959729060403396690

8665901 966892711 21 57069539883476944829695994006826658841 65436475088
71211 06401 6525637362495478239385828348965859163026 3

34795869855325290585531 0830980052290672903416198831 81 24941 8973861 34
9224255744520027150540959760966483403378566624107

55491 8896231 54451 3861 1372561 3554761 883229833964390153977827445730 41
4606221 134336894701 3989047559898986111 30692428602031

7420019784724489007329307894118535818861 57421 061 87737041 93962003878
4045953013076210561 31 778788078629517257839126825563

4453701 3072802406421 4980941 7461 059011 41 380676697987031 6253426481 744
654284751 532633303550368525548314170961 047089431 429

6069815931 9661 351 41 7674723225930736402295984910175032306340027424 98
2697394630621 9252779005640905488135012937033741 9430

41 7498535909982641 31 37396230771 7638386098788728884046404546578369 5
78524670895091 4943905489772232713061 601 211 592263360

4250189214175758350210766517529560741 8884931 971 208948830781 47535360
396716464435036089729557890373989462262430136933906

1 9838846448698056589530749599324274986868591 6303694986979443044532 8
0943048357819658258574012001 51 3522857797823381 00688

3761 982269000871 3259944451 351 441 61 4951 669854227880611 33086940453743
83581 70562382351 720675308291 0838925872648031 3552152

2983362862035497228856972667986459879505201 81 2331 653534893690915000
21553055481 641 471 9420469241 641 2485672248266749592 91

81 43705331 605949047751 21 8748602934091 43295421 446509063416471 0029303
01 31 231 81 61 9676825548986635651 4236744600237026528 75

1 2238967737956087308752720299366574620896328753523993673588704924 08
877491 3749816978399734734179596993930084386608009 40

El mensaje recuperado fue

¡YO NEZAHUALCOYOTL LO PREGUNTO ACASO DE VERAS SE VIVE CON RAIZ EN LA TIERRA NO PARA SIEMPRE EN LA TIERRA SOLO UN POCO AQUI AUNQUE SEA DE JADE SE QUIEBRA AUNQUE SEA DE ORO SE ROMPE AUNQUE SEA PLUMAJE DE QUETZAL SE DESGARRA NO PARA SIEMPRE EN LA TIERRA SOLO UN POCO AQUI PERCIBO LO SECRETO LO OCULTO OH VOSOTROS SENORES ASI SOMOS DE CUATRO EN CUATRO NOSOTROS LOS HOMBRES TODOS HABREMOS DE IRNOS TODOS HABREMOS DE MORIR EN LA TIERRA COMO UNA PINTURA NOS IREMOS BORRANDO COMO UNA FLOR NOS IREMOS SECANDO AQUI SOBRE LA TIERRA COMO VESTIDURA DE PLUMAJE DE AVE ZACUAN DE LA PRECIOSA AVE DE CUELLO DE HULE NOS IREMOS ACABANDO MEDITADLO SENORES AGUILAS Y TIGRES AUNQUE FUERAIS DE JADE AUNQUE FUERAIS DE ORO TAMBIEN ALLA IREIS AL LUGAR DE LOS DESCARNADOS TENDREMOS QUE DESAPARECER NADIE HABRA DE QUEDAR ESTOY EMBRIAGADO LLORO ME AFLIJO PIENSO DIGO EN MI INTERIOR LO ENCUENTRO SI YO NUNCA MURIERA SI NUNCA DESAPARECIERA ALLA DONDE NO HAY MUERTE ALLA DONDE ELLA ES CONQUISTADA QUE ALLA VAYA YO SI YO NUNCA MURIERA SI YO NUNCA DESAPARECIERA A DONDE IREMOS DONDE LA MUERTE NO EXISTE MAS POR ESTO VIVIRE LLORANDO QUE TU CORAZON SE ENDERECE AQUI NADIE VIVIRA PARA SIEMPRE AUN LOS PRINCIPES A MORIR VINIERON HAY INCINERAMIENTO DE GENTE QUE TU CORAZON SE ENDERECE AQUI NADIE VIVIRA PARA SIEMPRE !

Las pruebas fueron hechas en una **TELEVIDEO AT**, con palabra de 16 bits a una velocidad de 10 MHz, el tiempo aproximado del encriptamiento del mensaje fué de 3.20 min. y el de recuperación de 1.35 min. Cabe hacer notar que entre más grande sean los números primos utilizados y/o más grande sea el mensaje los algoritmos serán más lentos.

El diseño fué hecho para uso general, su funcionamiento no depende del tamaño de P y de Q, por tanto tampoco del tamaño de **N**, sólo el usuario debe definir su alfabeto, proponer los anteriores números, más el número D y su mensaje **M**.

Conclusiones

Mucho tiempo hubo que transcurrir para que se tuvuiera un importante avance en la prueba de primicidad para números enteros, aunque el algoritmo ideado por **APR** no es de una complejidad polinomial, su descubrimiento deja entrever que en un futuro no lejano y con el avance que en teoría de números se esta teniendo, se cuente con un algoritmo de dicha complejidad; por otra parte, los últimos estudios han dado evidencias de que es posible que no exista un algoritmo de complejidad polinomial para la factorización de números enteros, si esto fuera cierto, la seguridad del sistema criptográfico **RSA** estará asegurada por algún buen tiempo, ya que con los avances que se tienen en la fabricación de hardware hacen que los cálculos sean cada vez más rápidos, aminorando así el tiempo de máquina utilizado para la realización de cierta tarea. En el caso particular de la factorización de números enteros, **C. POMERANCE, J.W. SMITH y R. TULER** de la Universidad de Georgia, han publicado en este año el artículo " **A PIPELINE ARCHITECTURE FOR FACTORING LARGE INTEGERS WITH THE QUADRATIC SIEVE ALGORITHM** ", en el cuál describen la implementación del algoritmo de factorización conocido como "*criba cuadrática*" en un procesador diseñado especialmente para ello, han estimado el costo de su fabricación en aproximadamente $50, 000.00 dólares. Tal procesador permitirá la factorización de un número de 100 digitos en un tiempo de máquina equivalente a 2 semanas y un número de 144 digitos en un tiempo de un año; más sin embargo, el costo de factorizar a éste último es aproximadamente de diez millones de dólares.

Es con esta clase de avances de hardware que en el futuro la factorización de números enteros será cosa de algunas horas, no importando la existencia o no de un algoritmo de tiempo polinomial para ello, pero al igual que este desarrollo, la Criptografía avanza y muy seguramente dara una respuesta a ello.

Apéndice

A continuación se explica como pueden ser ejecutados los programas hechos en LISP, para ello es necesario contar con la versión μ-lisp *86* para computadoras personales (PC). Desde el sistema operativo cargar primero μ-lisp *86* con la siguiente instruccion, apareciento al último el prompt de μ-lisp **$**.

```
A>MULISP COMMON.LSP
```

El archivo COMMON.LSP contiene las funciones necesarias para ejecutar los siguientes programas: como primer paso debe cargarse a memoria todos los números primos que esten en el intervalo [3,70001; para ello ejecute lo siguiente desde μ-lisp :

```
$ (RDS B:LPRIM)
  INI
  NIL
$ (INI)
```

LPRIM es un archivo que consta solamente de la función INI que carga en una lista llamada **PRIMOS** todos los primos existentes en el intervalo mencionado, lo que aparece sin el símbolo **$** son las funciones de que consta el programa.

Para generar los números primos P, Q y D que intervendrán en el sistema **RSA**, hay que ejecutar lo siguiente .

```
$ (RDS B:GENERA2)
  GP
  VERIFICA
  PRIMOS_AUX
  PRIMOS_AUX1
  PRIMOS_AUX2
  BINARIO
  AP
  NIL
$ (GP M L)
```

El archivo GENERA2, contiene el programa del algortimo para la generación de números primos debido a O. Rabin. Aquí **M** y **L** son los números que se piden en la forma $N-1=2^{L}\cdot M$ para dicho algoritmo (ver pag. 44), automáticamente el programa genera el número **N**; si éste resulta ser un número compuesto, el programa genera otro número **N** diferente al anterior respetando la forma de **N-1**. En todo momento se emiten mensajes de lo que va sucediendo, si **N** es compuesto, el nuevo valor de **N** y si **N** es muy probablemente primo. De esta forma se generan **P**, **Q** y **D**, se deben guardar en un archivo y que ellos formen una lista, por ejemplo si **P=1949**, **Q=1993** y **D=2909**, el archivo debe contener exclusivamente lo siguiente :

(1949 1993 2909)

El cálculo del número **E** que cumpla $E*D\equiv 1 \pmod{(P-1)(Q-1)}$; se realiza como sigue :

```
$ (RDS B:PQDE)
  A
  MCD
  MCD1
$ (A)
```

El programa pedirá el nombre del archivo donde estan los números **P**, **Q** y **D**, dando como resultado el número **E**, el número **N=P*Q** es cálculado en este programa; se crean los siguientes archivos B:NE.DAT y B:ND.DAT, el primero de ellos contendrá a los números **N** y **E** para el encriptamiento de mensajes y el segundo los números **N** y **D** para el desencriptamiento de mensajes.

Para el encriptamiento de mensajes realice lo siguiente :

```
$ (RDS B:ENCRI)
  ENCRIPTAR
  TRANSFORMAR
  CONSTRUYE
  PARTICIONA
  ACOMPLETA
  EMPACA
  AUX-EMPACA
```

```
  ENCRIPTA
  AUX-ENCRIPTA
  BINARIO
  NIL
$ (ENCRIPTAR)
```

El programa inicialmente preguntará por el archivo donde reside el mensaje a encriptar, éste puede ser creado desde cualquier editor. debe iniciar con un paréntesis derecho (y terminar con un izquierdo), los carácteres que pueden formar el mensaje deben estar en (A, B, ...,Z,0,1, ...,9). Una vez leído dicho mensaje se pasa al procedimiento de encriptamiento emitiendo el mensaje : ">> ENCRIPTANDO ESPERE UN MOMENTO <<"; terminado éste, el programa preguntará el nombre del archivo donde residirá el mensaje encriptado, dado éste, se creará dicho archivo.

Para desencriptar mensajes :

```
$ (RDS B:DECRI)
  DESENCRIPTAR
  DESENCRIPTA
  AUX-DESENCRIPTA
  BINARIO
  EMITE-MENSAJE
  AUX-EMITE-MENSAJE
  NIL
$ (DESENCRIPTAR)
```

Inicialmente pide el nombre del archivo donde reside el mensaje encriptado, después inicia la ejecución de desencriptamiento emitiendo el mensaje ">> DESENCRIPTANDO ESPERE UN MOMENTO <<", terminado éste, pide el nombre del archivo donde residirá el mensaje desencriptado, creando un archivo.

Si después de haber sido generados los números **P**, **Q** y **D**, se desea someterlos al algoritmo de factorización "**p-1**"; haga lo siguiente :

```
$ (RDS B:PASO1)
  PASO1
  PRIM
  FACT
  COI
```

```
  ALGPOT
  BINARIO
  AP
  NIL
$ (PASO1 N A B1)
```

Lo anterior ejecuta el paso 1 del algoritmo, como entrada van, el número a factorizar **N** (que en este caso sería **P**, **Q** o **D**), **A** es un número primo relativo a **N** y la cota **B1** (ver pag. 18); si en un primer intento el programa da el valor **A=1**, es recomendable probar con nuevos valores tanto para **A** como para **B1**, en caso de persistir el valor **A=1**; se pasa a ejecutar el paso 2 de la siguiente forma :

```
$ (RDS B:PASO2)
  PASO2
  FORMASJ
  ENCUENTRA
  FSJ
  FORMADJ
  FORMABJ
  CALCULA
  MLP
  MULTIPLICA
  NIL
$ (PASO2 N RI C B1 B2)
```

Aquí **RI** es el último residuo calculado en el paso 1, éste se encuentra en la variable RI creada en el programa anterior, **C** es una constante entera mayora 1 (por lo general **C=2,3**), **B2** es la segunda cota (ver pag. 20). El programa va mostrando los valores que toma la variable **GTS** (que juega el papel de A_t), en caso de que A_t=**N** y A_{t-1}=1 se debe volver a intentar la factorización en un nuevo intervalo (B1,B2); para demostración de lo anterior hacemos referencia a los ejemplos expuestos en las páginas 20 y 21.

Bibliografía

1. L. M. Adleman, C. Pomerance y R. S. Rumely.
"On distinguishing prime numbers from composite numbers"
Ann. of Math., Vol. 117 (1983), 13-206.

2. L. M. Adleman, R. L. Rivest y A. Shamir.
"A method for obtaining digital signatures and public-key cryptosystems"
Comm. ACM, Vol. 21 (1978), 120-126.

3. J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman y S. S. Wagstaff Jr
"Factorizations of $b^n \pm 1$, b=2, 3, 5, 6, ,7, 10, 11, 12 up to high powers"
Contemporary Mathematics, Vol. 22, American Mathematic Society, Providence-Rhode Island, 1983.

4. H. Cohen y H. W. Lenstra Jr.
"Primality testing and Jacobi sums"
Math. Comput., Vol. 42 (1984), 297-330.

5. H. Cohen y A. K. Lenstra.
"Implementation of a new primality test"
Math. Comput., Vol. 48 (1987), 103-121.

6. W. Diffie y M. E. Hellman.
"New directions in Crytography"
IEEE Trans. Inform. Theory, IT-22 (1976), 644-654.

7. J. D. Dixon.
"Factorization and primality test"
Amer. Math. Monthly, Vol. 11 (1984), 333-352.

8. "EL FINANCIERO".
Periódico, 30 de diciembre de 1987.
México D. F., pag. 41.

9. E. Grosswald.
Topics from the theory of numbers
Birkhauser, Boston. 1984.

10. M. E. Hellman.
"An overview of public key cryptography"
IEEE Communications Society Magazine, (1978), 24-32.

11. M. E. Hellman y R. C. Merkle.
"Hiding information and signatures in trapdoor knapsacks"
IEEE Trans. Inform. Theory, IT-24 (1978), 525-530.

12. A. G. Konheim.
Cryptography : a primer.
John Wiley and Sons, New York, 1981.

13. A. Lempel.
"Cryptology in transition"
Computing Surveys, Vol. 11 (1979), 285-303.

14. G. L. Miller.
"Riemann's hypothesis and test for primality"
J. Comput. System Sci., Vol. 13 (1976), 300-317.

15. L. Monier.
"Evaluation and comparison of two efficient probabilistic primality testing algorithms"
Theoret. Comput. Sci., Vol. 12 (1984), 97-108.

16. J. M. Pollard.
"Theorems on factorization and primality testing"
Proc. Camb. Philos. Soc., Vol. 76 (1974), 521-528.

17. C. Pomerance.
"Recent developments in primality testing"
Math. Intelligencer, Vol. 3 (1981), 97-105.

18. C. Pomerance, J. W. Smith y R. Tuler.
"A pipeline architecture for factoring large integers with the quadratic sieve algorithm"
SIAM J. Comput., Vol. 17 (1988), 387-403.

19. M. O. Rabin.
"Probabilistic algorithm for testing primality"
J. Number Theory, Vol. 12 (1980), 128-138.

20. J. E. Shockley.
Introduction to number theory.
Holt, Rinehart and Winston, Inc., New York, 1967.

21. G. J. Simmons.
"Cryptology : The Mathematics of secure communication"
Math. Intelligencer, Vol. 1 (1979), 233-246.

22. R. Solovay y V. Strassen.
"A fast Monte-Carlo test for primality"
SIAM J. Comput. Vol. 6 (1977), 84-85.

23. R. E. Tarjan.
"Complexity of combinatorial algorithms"
SIAM Review. Vol. 20 (1978), 457-491.

24. I. M. Vinogradov.
Fundamentos de la teoría de los números
MIR. Moscú, 1977.

25. H. C. Williams.
"A modification of the RSA public-key encryption procedure"
IEEE Trans. Inform. Theory. IT-26 (1980), 726-729.

26. H. C. Williams.
"A p+1 method of factoring"
Math. Comput., Vol. 39 (1982), 225-234.

27. H. C. Williams.
"Factoring a computer"
Math. Intelligencer, Vol. 6 (1984), 29-36.

El jurado designado por la Sección de Computación del Departamento de Ingeniería Eléctrica del Centro de Invetigación y de Estudios Avanzados del I. P. N., aprobó esta tesis el día 18 de octubre de 1988.

Dr. Manuel E. Guzmán Rentería

Dr. Guillermo B. Morales Luna

Dr. Horacio Tapia Recillas