UCLV
Universidad Central
"Marta Abreu" de Las Villas

MFC
Facultad de Matemática
Física y Computación

Departamento de Matemática

TRABAJO DE DIPLOMA

Título: Creación de un procedimiento para atacar la seguridad del RSA.

Autor: Maria Esther Leyva Borges

Tutores: Dr. C. Lucía Argüelles Cortés

MSc. Gonzalo Palencia Fernández

Santa Clara, Julio 2019

UCLV
Universidad Central
"Marta Abreu" de Las Villas

MFC
Facultad de Matemática
Física y Computación

Math department

DIPLOMA THESIS

Title: Creation of a procedure to attack the safety of the RSA.

Author: Maria Esther Leyva Borges

Thesis Director: Dr. C. Lucía Argüelles Cortés

MSc. Gonzalo Palencia Fernández

Santa Clara, July 2019

RESUMEN

Hoy en día cuando se habla de criptoanálisis solo se piensa en personas dedicadas a espionaje o cualquier forma ilícita de obtención de información. Pero la realidad de esta ciencia es otra, el criptoanálisis es la ciencia inversa a la criptografía tal como la adición y la sustracción, no existe criptografía sin criptoanálisis. Todo criptosistema guarda su seguridad en un grupo de criptoanalistas dedicados a encontrar sus debilidades y así mejorarlas.

Este trabajo tiene como propósito crear un algoritmo criptoanalítico para atacar el criptosistema RSA basado en un novedoso algoritmo de factorización propuesto por la autora. Para ellos primero se crea el algoritmo de factorización, se crea además un algoritmo de obtención de claves privadas parejas, demostrando que la existencia de estas no afecta ni la efectividad ni la seguridad de este algoritmo, luego se crea un algoritmo para ya factorizada la clave pública y obtenida la privada descifrar un mensaje cifrado mediante RSA.

Se utilizó el software Python para la implementación de estos algoritmos debido a la complejidad de los cálculos.

SUMMARY

Nowadays, when we talk about cryptanalysis, we only think of people dedicated to espionage or any illicit way of obtaining information. But the reality of this science is another, cryptanalysis is the reverse science of cryptography such as addition and subtraction, there is no cryptography without cryptanalysis. Every cryptosystem keeps its security in a group of cryptanalysts dedicated to finding their weaknesses and thus improving them. The purpose of this work is to create a cryptanalytic algorithm to attack the RSA cryptosystem based on a novel factorization algorithm proposed by the author. For them, first, the factorization algorithm is created, and an algorithm to obtain matching private keys is also created, demonstrating that the existence of these does not affect the effectiveness or safety of this algorithm, then an algorithm is created to factor the key. public and obtained the private decrypt an encrypted message using RSA. Python software was used to implement these algorithms due to the complexity of the calculations.

DEDICATORIA

A mi mamita porque la amo,

A mi hermana por siempre estar ahí,

A mi abuelita Mary por apoyarme tanto

Y en especial a mi chiki, la princesa de mi vida

por ser mi motor impulsor.

AGRADECIMIENTOS

A la primera persona que quiero agradecer, o mejor dicho personita, es a Maradita, mi chiki que con sus abracitos, sus besitos y sus “cuando nos graduamos mema” hizo que nunca me rindiera. A mi mamita que con todo y sus resabios me ha dado fuerzas para seguir adelante y me ha enseñado que no debo rendirme hasta lograr mis sueños. A mi titi por regalarme lo más lindo que tengo en la vida, por cuidarme y por darme ánimos en los momentos que en que más los he necesitado. A mi papá por apoyarme todo este tiempo, a los jimaguas porque ha sido más que papás para mí, porque en cada momento que los he necesitado siempre han estado. A mi abuelita Mary por enseñarme a ser fuerte y no darme por vencida. A Chabe por siempre levantarme la autoestima con sus cosas. A Lili y a Rosy por cuidarme como su sobrina.

A Héctor por ser mi paño de lágrimas, por sus buenos consejos en los momentos en que más confundida he estado.

A Yamilitomorfa por cuidarme, apoyarme y molestarme estos cuatro años. A Dianerys, a Gianny, a los amigos que veo a diario y a los que hace rato no puedo ver.

Al profesor Morgado por ser el primero en apoyarnos con nuestras locuras, a la profesora Lucia por guiarme y por enseñarme tanto y a todos esos profesores que han contribuido a mi formación tanto académica como personal .

En fin a todos los que han hecho de mi sueño una realidad.

La autora

ÍNDICE

Introducción

Introducción.

El mundo se encuentra en la Era de la Información, a veces es necesario protegerla, otras veces es necesario poner al descubierto la que alguien más protegió. Del proceso de cifrar y descifrar la información se ocupa una de las ramas más antiguas de las Matemáticas, la criptología, la cual pretende ser el campo de acción en el que está situada esta investigación.

La criptología es la disciplina científica que estudia la escritura secreta, mensajes que al ser procesados se convierten en difíciles o imposibles de leer por entidades no autorizadas. Tiene una historia milenaria y sus inicios se pierden en el alba de la civilización. Se ha rastreado su origen en inscripciones funerarias egipcias en las que la escritura jeroglífica habitual era sustituida por otra diferente. El propósito no era propiamente criptográfico, sino una especie de juego o desafío al lector. Sin embargo, en la actualidad la criptología es más que un juego, con la aparición de los ordenadores su objetivo cambió y es hoy la encargada de estudiar las técnicas que se encargan de proporcionar seguridad a la información. La criptología está compuesta por cuatro disciplinas que son: (García, R. d. M., 2008)

• La esteganografía, se ocupa de ocultar mensajes con información privada por un canal inseguro, de modo que no sea ni siquiera percibido;

• El estegoanálisis, se ocupa de detectar mensajes ocultos con técnicas esteganográficas;

• La criptografía, se ocupa del estudio de algoritmos, protocolos y sistemas que se utilizan para proteger la información;

• Y el criptoanálisis, que es el campo de aplicación, en el que pretende la autora enmarcar sus resultados, se ocupa de conseguir el significado de mensajes construidos mediante criptografía sin tener autorización para ello.

Si se busca en internet posiblemente se encontraran miles de libros y artículos dedicados a la criptografía, mientras que del criptoanálisis se habla y se escribe muy poco, su condición de ser asociado a espionaje lo convierte en ese tema que nadie quiere tratar y del que nadie quiere escribir, para que se trataría este tema tan escambroso si se puede tratar otro tema muy parecido pero que

garantiza seguridad y protección, la criptografía. El criptoanálisis usualmente es asociado con espionaje o cualquier forma ilícita de obtención de información, pero la realidad es que el criptoanálisis tiene también otros fines.

En septiembre de 1918 los americanos llevaban meses preparando un ataque para destruir el saliente de Saint Mihiel cuando de repente se paralizaron los preparativos, no porque el objetivo hubiera perdido su valor estratégico, ni porque el enemigo se hubiera reforzado, sino simplemente porque un estudiante de criptografía recién llegado de Washington puso en conocimiento del cuartel general que había descifrado con relativa facilidad un mensaje secreto emitido por radio interceptado en el sector americano. Si un aficionado podía descifrar con facilidad un mensaje secreto, los expertos alemanes evidentemente también y por lo tanto debían de estar al tanto de los preparativos del ataque.

Y este precisamente es un ejemplo de la importancia del criptoanálisis. Detrás de cada grupo de criptógrafos hay un grupo de criptoanalistas dedicados a comprobar la seguridad de los algoritmos, asegurándose de que no puedan ser rotos por otros criptoanalistas.

Aunque de ello no se hable mucho, el estudio del criptoanálisis es de vital importancia para la seguridad de la información. En la actualidad, tal y como debe ser, la criptografía va mucho más avanzada que el criptoanálisis y no se ha podido construir ningún método que permita atacar cualquier mensaje cifrado, solo se descifran casos particulares.

Dentro del criptoanálisis existen dos ramas: el criptoanálisis activo, en el que el atacante se dedica a modificar la información que recibirá el destinatario y el criptoanálisis pasivo en el que el atacante solo conoce la información del mensaje, no le hace ninguna transformación y en el cual el atacante tiene pocas posibilidades de ser descubierto.

Debido a la gran relación entre la criptografía y el criptoanálisis, no se puede estudiar criptoanálisis sin un gran conocimiento sobre criptografía porque al igual que la suma y la resta, y la multiplicación y la división estas ciencias están conectadas siendo una la inversa de la otra.

Por esta gran relación, el estudio del uno está basado en el desarrollo de la otra. La criptografía es tan antigua como los escritos, y con ella el método simétrico, con él esta rama de la matemática se estructuró y formalizó, utilizando técnicas donde para cifrar y descifrar se utiliza la misma clave, las llamadas técnicas simétricas donde el atacante para descifrar un mensaje solo debía conocer la clave que poseía el emisor y el receptor. En el pasado siglo con el gran avance de las comunicaciones esta técnica dejó de ser óptima, pues la clave debería cambiar para cualesquiera dos elementos que se estuvieran comunicando y cuando la comunicación era de una entidad con millones de personas cuantas claves debería conocer la entidad y como identificar de quien era cada clave. En ese momento surgió la necesidad de una transformación en la criptografía y así surge la criptografía asimétrica, la criptografía en la que dos entidades entienden el mismo mensaje con claves diferentes. Teoría que fue presentada en 1976 por dos destacados científicos, Whitfield Diffie y Martin Hellman, que aunque no lograron crear un criptosistema, crearon un protocolo de intercambio de clave que sentó las bases teóricas para la nueva rama de la criptografía que iba naciendo, la criptografía asimétrica o de llave pública . Un año más tarde, en 1977, un equipo de tres científicos, R. Rivest, A. Shamir y L. Adleman, lograron crear el primer criptosistema de llave pública, al que llamaron RSA en honor a sus nombres. En el cual cada entidad o usuario publica una clave que se conoce como clave pública y mantiene en secreto otra llamada clave privada, para cifrar un mensaje cada usuario debe buscar la clave pública del destinatario y solo el destinatario podrá leer el mensaje descifrándolo con su clave privada. Esta nueva rama de la criptografía dejó al criptoanálisis atrás.

En la actualidad todavía no se ha encontrado una técnica de descifrado que permita conocer a un atacante cualquier mensaje cifrado con RSA, pues su seguridad está garantizada en la dificultad que representa la factorización de enteros, cuando los enteros son números grandes (más de 1024 bits).

Como cualquier matemático la autora presenta gran interés por la teoría de números, comenzando así durante la formación de pregrado una investigación enfocada en los números primos, donde se propone una formula generatriz para los llamados ´´ladrillos´´ con los que se construyen los números naturales.

Ya terminada la investigación, la conversación con expertos, tanto en teoría de números, en criptografía o en matemática en general, hizo nacer en la autora un nuevo interés, que no fue más que una aplicación a la teoría que ya había desarrollado. Estudiando la bibliografía descubrió la escasez de teoría sobre el criptoanálisis a pesar de su importancia para la seguridad, y nació así un nuevo objetivo investigativo: crear un algoritmo criptoanalista para el criptosistema RSA, pretendiendo sustituir el método prueba y error, por un cuerpo de conocimientos teóricamente justificados de criptoanálisis y con ello el mejoramiento de los métodos criptográficos.

Para ello es necesario analizar el criptosistema RSA, determinar sus debilidades, atacar las mismas, crear un algoritmo que permita descifrar un mensaje cifrado con este método y mostrar ejemplos que evidencien la vulnerabilidad del mismo.

Por lo anterior, se plantearon las siguientes interrogantes científicas:

- ¿Cómo funciona el criptosistema RSA?
- ¿Cuáles son sus principales debilidades?
- ¿Existirá un algoritmo capaz de atacar estas debilidades y así romper la seguridad del criptosistema?
- ¿Será posible, dado un mensaje cifrado, conocer el texto en claro mediante este algoritmo de ataque?

El trabajo consta de tres capítulos. El capítulo I titulado “Fundamentos teóricos” ofrece una base para la comprensión de cada una de las temáticas que se necesitarán durante el trabajo, ofreciendo definiciones, caracterizaciones, métodos de solución así como un pequeño bosquejo histórico sobre el tema del que se trata el trabajo. El segundo capítulo “Construcción del algoritmo de ataque” como bien dice su nombre se dedica a construir el algoritmo, comenzando por explicar cómo funciona el criptosistema RSA, para luego definir dónde reside su seguridad y cuáles son los puntos débiles que se podrán atacar. A partir de sus debilidades y con la base teórica ya definida se propone el algoritmo de ataque y sus fundamentos teóricos, mostrando con ejemplos sencillos como funciona este algoritmo. Debido a la complejidad de

los cálculos a realizar es necesario la utilización de un software, en este caso se utiliza el *Python*. Y es precisamente la implementación de este algoritmo en *Python* lo que da lugar al tercer capítulo titulado “Implementación computacional del algoritmo de ataque” en el que se explica primeramente por qué se escoge este software, haciendo énfasis en sus facilidades y posibilidades y mostrando además las funciones implementadas, así como ejemplos con un grado de complejidad superior a los mostrados en el capítulo II.

A partir de la creación por la autora de un algoritmo de factorización de enteros, quedó en evidencia la vulnerabilidad del algoritmo RSA cuando existen maneras menos complejas de factorizar enteros. De ahí se plantea el siguiente problema científico:

- ¿Es posible mejorar el algoritmo RSA a partir del uso de maneras más sencillas de la factorización de enteros?

El aporte teórico de la investigación viene dado a partir de mostrar debilidades del algoritmo RSA para desde ahí poder mejorar su funcionamiento en futuras investigaciones.

Métodos del nivel teórico

- **Análisis-síntesis**

  Al realizar la revisión bibliográfica, para reconocer las características principales del criptoanálisis, la criptografía, el método RSA y la factorización de enteros.

- **Inducción-deducción**

  Posibilitó realizar inferencias y deducciones de los principales sustentos teóricos que fundamentan la investigación, permitiendo la creación de nuevas ideas que condujeron a la solución de la problemática presentada, como fue la posible utilización de la factorización de enteros en el ataque del criptosistema de llave pública RSA.

- **Histórico-lógico**

  Se utilizó para ver la evolución del tema, principalmente en el proceso de evolución de la criptografía y el uso de los números primos en ella, viendo así la necesidad de utilizar nuevas teorías matemáticas en el tema objeto de estudio para atacar de manera novedosa el RSA.

Métodos del nivel empírico

- **Análisis documental**

  Este método fue considerado para la revisión bibliográfica de un grupo relevante de documentos relacionados con el tema, el cual incluye artículos, libros, tesis, datos de la práctica y sitios de Internet. Los documentos más valiosos para la configuración del trabajo han sido:

  - Una simulación simplificada de la criptografía, Alicia Ayuso Solís, Paula Goicoechea Núñez, *et al.,* 2009.
  - Modern Cryptography: Theory and Practice, Wembo Mao Hewlett, Prentice Hall PTR, 2003
  - Criptoanálisis más utilizados en la actualidad, Enrique Sánchez Acosta, 2012.

El valor metodológico que posee la investigación viene dado a partir del desarrollo de ejemplos básicos que ilustran los principales conceptos y muestran paso a paso cómo atacar el algoritmo RSA.

Capítulo 1

Capítulo I: Fundamentos teóricos.

Las matemáticas como ciencia siempre han tenido gran penetración en todas las áreas de desarrollo de la humanidad, a veces de manera indirecta, otras de manera más inmediata, desde sus campos más abstractos como la teoría de números, la geometría algebraica, la topología, el análisis real y complejo, hasta los más aplicados, como la probabilidad, la estadística, las ecuaciones diferenciales, todas tienen una aplicación a diferentes áreas de la tecnología, y es imposible desligar a las matemáticas del desarrollo actual en todo el mundo. Dentro de la teoría de números se le presta especial atención a los números primos, debido a que pueden considerarse como los «ladrillos» de las matemáticas, los átomos de la aritmética o el código genético de los números.

1.1 Números primos.

Un número primo es, por definición, un entero positivo mayor que 1 que es divisible, solamente, por sí mismo y la unidad. Además, todo entero positivo mayor que 1 no primo puede ser escrito de forma única como el producto de primos. (Ayuso, A. 2009)

Los primeros intentos de descifrar los misterios de los números primos surgieron en la antigua Grecia. Allí fue donde se establecieron los principios matemáticos sobre los que se ha trabajado desde entonces. Los pitagóricos se encargaron de profundizar en los conceptos fundamentales de la aritmética, otorgándole a los números un carácter casi místico. Fueron ellos quienes comenzaron a multiplicar y operar con los números, dándose cuenta de que existían algunos de ellos que eran imposibles de reducir. Los números primos de la forma $4n + 1$ son conocidos como los números primos pitagóricos, de ellos se hicieron largas listas tratando de encontrar el último.

Hasta el año 300 a.c, cuando se publica el libro Los Elementos de Euclides, se pensaba que los números primos eran finitos. Pero en este libro aparece una ingeniosa demostración que dice lo contrario. (Sánchez, J. M. 2011)

Una vez conocida la infinitud del conjunto de los números primos, el siguiente paso sería cuantificar la cantidad de números primos que existen menores que

un número dado n, la función que devuelve este resultado se conoce como $\pi(n)$. La herramienta más útil que se conoce para realizar este proceso es la Criba de Eratóstenes. Este es un algoritmo utilizado para eliminar los números compuestos entre 1 y n, y se basa en la siguiente observación: si los enteros $x \leq n$ no son divisible por ningún primo menor o igual que $\sqrt{n}$, entonces x es primo.

Cuando n es pequeño la criba de Eratóstenes permite calcular $\pi(n)$, pero cuando n crece la complejidad de la criba es muy alta. Entre el siglo XVIII y XIV Adrien Marie Legendre y Carl Friedrich Gauss trabajaron de manera independiente en encontrar $\pi(n)$ llegando a un resultado que aunque no es exacto es bastante aproximado para números grandes,

$$\pi(x) = \frac{x}{\log x}$$

Otro de los grandes misterios de la matemática cuando a números primos se refiere es la existencia de una fórmula capaz de generar exactamente toda la sucesión de los números primos.

Muchos de los grandes matemáticos de la historia han dedicado parte de su estudio a encontrar esta fórmula. En el siglo XVII el francés Pierre de Fermat se destaca en las investigaciones en la aritmética, y aunque no era matemático de profesión sino un hombre de leyes, hizo grandes aportes a la matemática, entre ellos conjeturó que todos los números de la forma $2^{2^n} + 1$ eran primos y lo comprobó hasta $n = 4$. Años más tardes el suizo Leonhard Euler, a quien se le conoce como el matemático más prolífero de la historia, demostró que para $n = 5$ se generaba un número compuesto y hasta nuestros días no se ha podido encontrar ningún otro número de Fermat que sea primo, además de los que él conocía. Otro de los grandes aportes del francés fue lo que se conoce como Pequeño Teorema de Fermat, que plantea:

Si p es un número primo, y a un número natural mayor que 1 entonces $a^p - a$ es divisible por p.

Otro grande en la historia de las matemáticas del siglo XVII que también investigó este gran misterio fue el monje, también francés, Marin Mersenne quien proponía que para p primo $2^p - 1$ tiene grandes probabilidades de ser

primo, aun en la actualidad se buscan números primos con esta forma y son los conocidos primos de Mersenne.

1.2 Ecuaciones diofánticas.

Una ecuación $f(x_1, x_2, \dots, x_n) = b$ es llamada ecuación diofántica si y solo si $f(x_1, x_2, \dots, x_n)$ es un polinomio en $x_1, x_2, \dots, x_n$ con coeficientes enteros, b constante tal que $b \in \mathrm{Z}$ y solo son de interés las soluciones enteras. (I. N. Bronshtein, K. A. S., G.Musiol, H.Muehling, 2005) El nombre diofántica viene del matemático Diofanto de Alejandría, conocido como el padre del álgebra.

1.2.1 Ecuaciones diofánticas lineales

Las ecuaciones diofánticas lineales de dos incógnitas tienen la forma $ax + by = c$ donde $a, b, c \in \mathrm{Z}$ tal que $(a, b) \neq (0,0)$.

Teorema 1.2.1.1: La ecuación diofántica $ax + by = c$ tiene solución si y solo si d es múltiplo de c, donde $d = mcd(a, b)$. (Titu Andreescu, D. A., Ion Cucurezeanu, 2010)

1.2.2 Ecuaciones diofánticas de Pell

Euler atribuyó los primeros estudios serios de las soluciones no triviales de la ecuación de la forma $u^2 - Dv^2 = 1$ a John Pell. Aunque se descubrió que Pell nunca había considerado el estudio de estas ecuaciones, y a quien se le debía asignar este mérito era a Fermat, pasaron a la historia como las ecuaciones de Pell.

Ya conocida la solución trivial de las ecuaciones de este tipo, $u = 1, v = 0$, se necesitan condiciones para encontrar las soluciones no triviales.

Teorema 1.2.2.1: Si D es un entero positivo, no es un cuadrado perfecto entonces la ecuación $u^2 - Dv^2 = 1$ tiene infinitas soluciones naturales y la forma general de la solución está dada por $(u_n, v_n)\ con\ n \geq 0$ donde $u_{n+1} = u_1u_n + Dv_1v_n\ , v_{n+1} = v_1u_n + u_1v_n$ donde (u_1, v_1) es la solución trivial. (Titu Andreescu, D. A., Ion Cucurezeanu, 2010)

1.2.3 Ecuaciones diofánticas cuadráticas

La ecuación diofántica cuadrática tiene la forma $ax^2 + bxy + cy^2 + dx + ey + f = 0$.

Esta ecuación representa un cono en el plano cartesiano, se desea reducirla hasta obtener una forma canónica. Se introduce el discriminante de la ecuación $\Delta = b^2 - 4ac$.

- Si $\Delta < 0$ el cono define una elipse.
- Si $\Delta = 0$ el cono define una parábola.
- Si $\Delta > 0$ el cono define una hipérbola.

Este tercer caso será al que se le prestará mayor atención.

Haciendo una serie de transformaciones esta ecuación se puede reducir a la forma general de una ecuación de Pell $u^2 - Dv^2 = N$

Ejemplo:

$2x^2 - 6xy + 3y^2 + 1 = 0$ Donde $a = 2, b = -6, c = 3, d = 0, e = 0, f = 1$

$$\Delta = b^2 - 4ac = (-6)^2 - 4 * 2 * 3 = 12 > 0$$

Esta ecuación se puede escribir como $u^2 - 3v^2 = 1$ donde $u = x, v = y - x$

1.3 Factorización de enteros.

Otro de los grandes problemas de las matemáticas es dado un número entero encontrar sus factores primos. Descomponer dos números de igual longitud no tiene por qué tener la misma complicación; sin embargo, actualmente se considera que los casos más difíciles son aquellos para los que los factores son dos números primos, elegidos al azar, de aproximadamente el mismo tamaño.

Por el teorema fundamental de la aritmética, cada entero positivo no primo tiene una única descomposición en números primos (factores primos). La

mayor parte de los algoritmos de factorización elementales son de propósito general, es decir, permiten descomponer cualquier número introducido, y solo se diferencian sustancialmente en el tiempo de ejecución.

$$\left(\begin{array}{c|c} 72 & 2 \\ 36 & 2 \\ 18 & 2 \\ 9 & 3 \\ 3 & 3 \\ 1 & \end{array}\right) \Leftrightarrow 72 = 2^3 * 3^2$$

Si un número grande, de b bits es el producto de dos primos de aproximadamente el mismo tamaño, no existe algoritmo conocido capaz de factorizarlo en tiempo polinómico. Esto significa que ningún algoritmo conocido puede factorizar en tiempo $O(b^k)$, para cualquier constante k, aunque existen algoritmos que son más rápidos que $O(a^b)$ para cualquier $a > 1$. En otras palabras, los mejores algoritmos son súper-polinomiales, pero sub-exponenciales.

Algunos de los algoritmos más conocidos para la factorización de enteros son:

- El método de factorización de Dixon (conocido también como método de los cuadrados aleatorios de Dixon o algoritmo de Dixon); es el método prototípico de base de factores, y el único método de este tipo para el cual los límites de ejecución no se basan en conjeturas sobre las propiedades de suavidad de los valores de un polinomio conocido.
- La factorización con fracciones continuas, conocido como método de factorización con fracciones continuas (CFRAC del inglés Continued Fraction Factorization Method). Es un algoritmo de propósito general, lo que significa que es apropiado para factorizar cualquier entero n, no dependiente de una forma espacial o de propiedades. Fue descrito por D. H. Lehmer y R. E. Powers en 1931, y desarrollado como un algoritmo de computadora por Michael A. Morrison y John Brillhart en 1975.
- El algoritmo de criba cuadrática (QS del inglés quadratic sieve), en la práctica, el segundo método más rápido conocido (después de la criba general del cuerpo de números naturales). Es todavía el más rápido para enteros que tienen 100 o menos dígitos decimales, y es considerado

mucho más sencillo que la criba de cuerpos numéricos. Es un algoritmo de factorización de propósito general, lo que significa que su tiempo de ejecución únicamente depende del tamaño del entero a ser factorizado, y no sobre una estructura especial o propiedades. Fue inventado por Carl Pomerance en 1981 como una mejora a la criba lineal de Schroeppel.

- La criba racional es esencialmente un caso especial de la criba general del cuerpo de números, y mientras que es mucho menos eficiente que el algoritmo general, es conceptualmente más simple. Así que, mientras es más bien un algoritmo de factorización sin utilidad en la práctica, es de utilidad como primer paso para aquellos que tratan de entender cómo funciona la criba general de cuerpo de números naturales.

1.4 Una nueva forma de factorizar enteros.

Se sabe que los números primos mayores que 2 están contenidos dentro de la sucesión de los impares, no múltiplos de 3.

Si cada número primo mayor que 3 se divide por 3, se obtiene un número de la forma $\frac{p}{3} = \begin{cases} n + \frac{1}{3} \ \ si\ n\ es\ par \\ n + \frac{2}{3} \ \ si\ n\ es\ impar \end{cases}$;

Proposición 1.4.1: Sea la sucesión construida de la siguiente forma: $p_n = \begin{cases} 3n + 1, \ \ si\ n \in \mathbb{N}\ y\ es\ par \\ 3n + 2, \ \ si\ n \in \mathbb{N}\ y\ es\ impar \end{cases}$ se puede probar que esta sucesión equivalente a la sucesión de los números impares mayores que 1 que no son divisibles por 3.

Demostración:

Si se denota I_n la sucesión de los números impares mayores que 1 que no son divisibles por 3.

Caso 1: Sea $n \in \mathbb{N}\ y\ es\ par$ entonces $n = 2k;\ k \in \mathbb{N}\ p_n = 3n + 1 = 3(2k) + 1 = 2(3k + 1) - 1 \Longrightarrow p_n\ es\ impar$ luego como $p_n = 3n + 1 \Longrightarrow p_n \equiv 1\ (mod[3])$ lo que es equivalente a que p_n no es divisible por 3.

Caso 2: Sea $n \in \mathbb{N}\ y\ es\ impar$ entonces $n = 2k-1;\ k \in \mathbb{N}\ \ p_n = 3n+1 = 3(2k-1)+2 = 2(3k)-1 \Longrightarrow p_n\ es\ impar$ luego como $p_n = 3n+2 \Longrightarrow p_n \equiv 2\ (mod[3])$ lo que es equivalente a que p_n no es divisible por 3.

Hasta ahora se ha probado que $\{p_n\} \subset I_n$

Sea $q \in I_n \Longrightarrow (q = 2k-1;\ k \in \mathbb{N}\ ) \wedge (q \equiv 1\ mod[3]\ \vee\ q \equiv 2\ mod[3]\ )$

Caso 1: Sea $(q = 2k-1;\ k \in \mathbb{N}\ ) \wedge (q \equiv 1\ mod[3]\ ) \Longrightarrow \exists\ m \in \mathbb{N}, tal\ que\ q = 3m+1$ se necesita probar que m es par, supongamos lo contrario, m impar, entonces $m = 2k-1;\ k \in \mathbb{N} \Longrightarrow q = 3m+1 = 3(2k-1)+1 = 6k-2 = 2(3k-1) \Longrightarrow q\ es\ par$ lo cual es un absurdo pues partimos de la hipótesis de que q es impar.

Caso 2: Sea $(q = 2k-1;\ k \in \mathbb{N}\ ) \wedge (q \equiv 2\ mod[3]\ ) \Longrightarrow \exists\ m \in \mathbb{N}, tal\ que\ q = 3m+2$ se necesita probar que m es impar, supongamos lo contrario, m par, entonces $m = 2k;\ k \in \mathbb{N} \Longrightarrow q = 3m+2 = 3(2k)+2 = 2(3k+1) \Longrightarrow q\ es\ par$ lo cual es un absurdo pues partimos de la hipótesis de que q es impar.

Se ha demostrado entonces que $I_n \subset \{p_n\}$ y a su vez $\{p_n\} \subset I_n$ por lo que $\{p_n\} = I_n$

Se observa que la sucesión de los números primos mayores que 3 está contenida en la sucesión I_n de los números impares mayores que 1 que no son divisibles por 3, y esta se ha probado que es equivalente a $p_n = \begin{cases} 3n+1,\ si\ n \in \mathbb{N}\ y\ es\ par \\ 3n+2,\ si\ n \in \mathbb{N}\ y\ es\ impa \end{cases}$ la cual se puede escribir como $p_n = \frac{6n+3+(-1)^{n+1}}{2}$ de modo que se puede proponer el siguiente teorema.

Teorema 1.4.1: Para todo número primo p mayor que 3 existe un natural n, tal que p se puede escribir de forma única como $p = \frac{6n+3+(-1)^{n+1}}{2}$.

Ahora, todo número escrito de esta forma no necesariamente va a ser primo. Se supone que n no es primo entonces va a ser múltiplo de 2 o de 3 o de números primos mayores que 3.

Si es múltiplo de 2 o de 3 es muy fácil verificarlo. En caso de que no ocurra esto, entonces será múltiplo de primos mayores que 3.

Se analizará para el caso $p_n = p_{n_1} * p_{n_2}$ donde p_{n_1} y p_{n_2} son primos, para otros casos basta con utilizar recursividad.

$$p_n = p_{n_1} * p_{n_2} = \frac{6n_1 + 3 + (-1)^{n_1+1}}{2} * \frac{6n_2 + 3 + (-1)^{n_2+1}}{2}$$

$$= \begin{cases} n\ par \begin{cases} n_1, n_2\ pares\ \ p_n = (3n_1+1)(3n_2+1) = 3(3n_1n_2 + n_1 + n_2) + 1, \\ n_1, n_2\ impares\ \ p_n = (3n_1+2)(3n_2+2) = 3(3n_1n_2 + 2n_1 + 2n_2 + 1) + 1, \end{cases} \\ n\ impar \begin{cases} n_1\ par, n_2\ impar\ p_n = (3n_1+1)(3n_2+2) = 3(3n_1n_2 + 2n_1 + n_2) + 2 \\ n_1\ impar, n_2\ par\ p_n = (3n_1+2)(3n_2+1) = 3(3n_1n_2 + n_1 + 2n_2) + 2 \end{cases} \end{cases}$$

Sea $p_n = \frac{6n+3+(-1)^{n+1}}{2} = \begin{cases} 3n+1\ \ si\ n\ es\ par \\ 3n+2\ si\ n\ es\ impar \end{cases}$ el número que se desea factorizar

Caso 1

Sea n par, existen dos posibilidades:

a-) n_1 y n_2 pares, se tiene que $3n + 1 = (3n_1 + 1)(3n_2 + 1) = 3(3n_1n_2 + n_1 + n_2) + 1$, de donde se obtiene que $n = 3n_1n_2 + n_1 + n_2$. Es decir n_1 y n_2 son solución de la ecuación diofántica $3xy + x + y = n$. Por tanto los dos factores que al multiplicarlos obtenemos p_n son $p_{n_1} = 3n_1 + 1$ y p_{n_2}=$3n_2 + 1$.

b-) n_1 y n_2 impares, se tiene que $3n + 1 = (3n_1 + 2)(3n_2 + 2) = 3(3n_1n_2 + 2n_1 + 2n_2 + 1) + 1$, de donde se obtiene que $n = 3n_1n_2 + 2n_1 + 2n_2 + 1$. Es decir n_1 y n_2 son solución de la ecuación diofántica $3xy + 2x + 2y + 1 = n$. Por tanto los dos factores que al multiplicarlos obtenemos p_n son $p_{n_1} = 3n_1 + 2$ y p_{n_2}=$3n_2 + 2$.

Caso 2

Sea n impar existen dos posibilidades que se reducen a una sola, n_1 y n_2 par e impar respectivamente, se tiene que $3n + 2 = (3n_1 + 1)(3n_2 + 2) = 3(3n_1n_2 + 2n_1 + n_2) + 2$ de donde se obtiene que $n = 3n_1n_2 + 2n_1 + n_2$. . Es

decir n_1 y n_2 son solución de la ecuación diofántica $3xy + 2x + y = n$. Por tanto los dos factores que al multiplicarlos obtenemos p_n son $p_{n_1} = 3n_1 + 1$ y p_{n_2}=$3n_2 + 2$.

En resumen, se ha obtenido que si p_n es compuesto entonces para buscar sus factores primos es suficiente encontrar la solución de las ecuaciones diofánticas cuadráticas, las cuales definen hipérbolas en el plano cartesiano:

- $3xy + x + y = n$,
- $3xy + 2x + 2y + 1 = n$
- $3xy + x + y = n$

1.5 Función de Euler.

Definición 1.5.1:

Dos números enteros son coprimos o primos entre sí si su máximo común divisor (mcd) es1, lo que es equivalente a decir que la fracción formada por ellos es irreducible. (Ayuso, A., 2009)

Definición 1.5.2:

Para todo $n \in \mathrm{N}, \mathrm{n} \geq 1$ la función representada por $\phi(n)$ y denominada función de Euler es el número de valores

$k \in \mathrm{N}: 0 \leq k < n$ y $mcd(k, n) = 1$ (Mao, W., 2003)

Propiedades de la función ϕ de Euler:

i. $\phi(1) = 1$
ii. Si p es primo $\phi(p) = p - 1$
iii. Si $mcd(m, n) = 1$ entonces $\phi(mn) = \phi(m)\ \phi(n)$
iv. Si $n = {p_1}^{e_1}.{p_2}^{e_2}.{p_3}^{e_3}.{p_4}^{e_4}.{p_5}^{e_5}$ ... es la descomposición en factores primos del número n entonces $\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\left(1 - \frac{1}{p_3}\right)$...

Teorema 1.5.1:

Este teorema es conocido como Teorema de Euler.

Si a y n son enteros coprimos, entonces n divide al entero $a^{\phi(n)} - 1$

Ejemplos:

- $\phi(7) = 7 - 1 = 6$ porque 7 es primo.
- $\phi(343) = \phi(7^3)$ como 7 es primo, 343 es coprimo con todos los menores a él excepto los múltiplos de 7.

$343 = 7^3 = 7 * 7^2$ Por tanto existen 7^2 múltiplos de 7 menores o iguales que 7^3. Luego existen $7^3 - 7^2$ números coprimos con 7^3 y $7^3 - 7^2 = 7^2(7 - 1) = 294$.

- Analizando el caso del **epígrafe 1.3** $72 = 2^3 * 3^2 \Leftrightarrow \phi(72) = \phi(2^3)\phi(3^2) = 72\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{3}\right) = 24$ por propiedad iii. y iv.

1.6 Aritmética modular.

La aritmética modular es un sistema aritmético para clases de equivalencia de números enteros llamadas clases de congruencia. Puede ser construida matemáticamente mediante la relación de congruencia entre enteros, es decir, que dos números naturales a y b tienen el mismo resto al dividirlos por un número natural m, llamado el $módulo$; se utiliza la notación $a \equiv bmod(m)$.

Sea $a \epsilon Z$, su clase de equivalencia en $mod(m)$ es el conjunto definido por $\bar{a} = \{x \epsilon Z : x \equiv a\ mod(m)\}$, donde a es el representante dela clase. En cada grupo Z_m existen m clases de equivalencia que van desde la clase del 0 hasta la clase $m - 1$.

1.6.1 Inversos.

Se requiere para su aplicación posterior generalizar en este contexto la idea de inverso de un número, que es otro que al multiplicarlo por el primero se obtiene como resultado la unidad.

Definición 1.6.1

El multiplicador modular inverso de un entero x módulo n es un entero a tal que $x^{-1} \equiv a\ mod(n)$, lo que es equivalente a decir $ax \equiv 1 mod(n)$.

Teorema 1.6.1:

El multiplicador inverso de x módulo n existe si y solo si x y n son coprimos, es decir, si $mcd(x,n) = 1$. (Ayuso, A., 2009)

Para calcular inversos se pueden seguir distintos caminos, uno de ellos es utilizando ecuaciones diofánticas lineales.

Dados x y n se supone que existe a tal que se cumple $ax \equiv 1 mod(n)$, por el teorema 1.6.1 se cumple que $mcd(x,n) = 1$.

De aquí que:

$ax \equiv 1 mod(n) \Leftrightarrow ax = kn + 1 \Leftrightarrow ax - kn = 1$

Resolviendo esta ecuación diofántica lineal cuyas variables son a y k se obtiene el inverso de x módulo n

Ejemplo:

$a * 5 \equiv 1 mod(17) \Leftrightarrow a * 5 = k * 17 + 1 \Leftrightarrow a * 5 - k * 17 = 1$

Solución : $a = 7$ y $k = 2$, por tanto 7 es el inverso modular de $5 mod(17)$.

1.7 Criptología

La criptología es la ciencia que estudia los criptosistemas, o también conocidos como los sistemas criptográficos o códigos secretos, es decir, es la disciplina científica que estudia la escritura secreta, mensajes que, al ser procesados de cierta manera, se convierten en difíciles o imposibles de leer por entidades no autorizadas. Dentro de esta ciencia se distinguen cuatro ramas fundamentales que son:

- La esteganografía: son todos aquellos métodos tendentes a la ocultación del mensaje en sí mismo. Forman parte de este grupo las tintas

simpáticas, la ocultación de bits e información en imágenes y las técnicas de los micropuntos,

- El estegoanálisis: son los métodos que se dedican a encontrar los mensajes que han sido ocultados mediante estegonagrafía,
- La criptografía: encargada de diseñar métodos para ocultar el contenido de un mensaje, no el mensaje en sí mismo. Para ello realiza toda una serie de transformaciones en el mismo de forma que sin conocer el origen y desarrollo de estas, así como una serie de información complementarias, en general la clave, sea prácticamente imposible conocer el contenido del mensaje original ,
- Y el criptoanálisis: son todos aquellos métodos dedicados a obtener el contenido del mensaje original sin conocimiento de parte de la información requerida para ello, principalmente la clave.

1.8 Criptoanálisis.

Usualmente al criptoanálisis se le asocia con temas tan escabrosos como espionaje, tanto militar como industrial o diplomático, y en general cualquier forma ilícita de obtención de información. Sería ingenuo negar estar cuestiones, sin embargo también es muy ingenuo el hecho de considerar el criptoanálisis como un juego de espías. Hay una razón básica para desechar esta idea, la razón económica romper un sistema es costoso en tiempo y material, se necesita en general gran cantidad de texto cifrado y conocer el destinatario para intentar obtener la clave, incluso conociendo todo lo anterior no se puede garantizar el éxito del atacante.

El criptoanálisis tiene dos objetivos esenciales. El primero de ellos y más importante es poder garantizar que un sistema criptográfico no presenta fallos. Y el otro es el descifrado de datos sensibles utilizados por elementos fuera de la ley para dañar la integridad de una entidad.

1.8.1 Criptoanálisis clásico.

Aproximadamente en el siglo IX, el matemático árabe YusufYaqubibnIshaq al-Sabbah Al-Kindi escribió un pequeño manuscrito para descifrar mensajes criptográficos. Así comenzó una de las técnicas clásicas del criptoanálisis:

a) El análisis de frecuencia o de sustitución:

Se trata del estudio e las frecuencias de las letras o los grupos de letras en un texto cifrado. Está basado en el hecho de que, dado en texto, ciertas letras o combinaciones de letras aparecen más a menudo que otras o combinaciones de letras aparecen más a menudos que otras, existiendo distintas frecuencias para ellas.

b) Método Kasiski ideado en 1863 para romper el sistema criptográfico Vigenere se basa en que ciertas secuencias de letras aparecen muchas veces en los textos originales, y consecuentemente los textos cifrados con Virgenere tiene las correspondientes secuencias de letras cifradas también muy repetidas. Por tanto, la distancia entre las cadenas más repetidas en el texto cifrado debe ser un múltiplo de la longitud de la clave K, y un buen candidato de longitud será el producto de los factores más repetidos en las distancias encontradas.

c) Índice de coincidencia: es un método desarrollado por William Friendman, en 1920, para atacar cifrados de sustitución polialfabética con claves periódicas. La idea se fundamenta en analizar la variación de las frecuencias relativas de cada letra, respecto a una distribución uniforme.

d) Telegrama Zimmermann (Primera guerra mundial): el telegrama Zimmermann fue un telegrama enviado por Arthur Zimmermann, ministro de Asuntos Exteriores del Imperio Alemán, el 16 de enero de 1917 (durante la Primera Guerra Mundial), a su embajador en México, conde Heinrich von Eckardt. En el mismo se instruía al embajador para que acercara al Gobierno mexicano una propuesta para formar una alianza contra los Estados Unidos. Fue interceptado por los servicios británicos de espionaje, y su contenido aceleró la entrada de los Estados Unidos en la guerra. El *Telegrama Zimmermann* fue interceptado y descifrado lo suficiente como para poder leer un esbozo

de su contenido, por los criptógrafos Nigel de Grey y William Montgomery de la unidad de la Inteligencia Naval Británica conocida como "*Habitación 40*" (en inglés *Room 40*), a cargo del almirante William R. Hall. Esto fue posible porque el código de cifrado utilizado por el Ministerio de Asuntos Exteriores de Alemania (llamado 0075) había sido analizado y parcialmente descifrado, se cree que utilizando mensajes ya descifrados y un libro de códigos anterior capturado por Wilhelm Wassmus, un agente alemán que trabajaba en el Oriente Medio.

e) Máquina Enigma (Segunda guerra mundial): Enigma era el nombre de una máquina que disponía de un mecanismo de cifrado *rotatorio*, que permitía usarla tanto para cifrar como para descifrar mensajes. Varios de sus modelos fueron muy utilizados en Europa desde inicios de los años 1920. Su fama se debe a haber sido adoptada por las fuerzas militares de Alemania desde 1930. Su facilidad de manejo y supuesta inviolabilidad fueron las principales razones para su amplio uso. Su sistema de cifrado fue finalmente descubierto y la lectura de la información que contenían los mensajes supuestamente protegidos es considerado, a veces, como la causa de haber podido concluir la Segunda Guerra Mundial al menos dos años antes de lo que hubiera acaecido sin su descifrado.

1.8.2 Criptoanálisis moderno.

El criptoanálisis moderno comienza por el dispositivo Bombe, que era un dispositivo electromecánico usado por los criptólogos británicos para ayudar a descifrar las señales cifradas por la máquina alemana Enigma durante la Segunda Guerra Mundial. La Armada y el Ejército de los Estados Unidos produjeron máquinas con la misma especificación funcional, pero diseñadas de una manera diferente.

La criptografía moderna se ha vuelto más impenetrable al criptoanalista que los métodos de pluma y papel del pasado, y parece que en la actualidad llevan ventaja sobre los métodos del puro criptoanálisis.

Actualmente existen dos tipos de ataque criptoanalíticos. El primero es el ataque activo en el que el atacante modifica el flujo de datos o crea flujos falsos. En este tipo de ataques se usan varias técnicas como:

- La suplantación: el atacante trata de suplantar la identidad de la persona que tiene la autorización, para que el sistema recepcione que es él y le de alguna información sobre la clave, aunque esta esté cifrada y poder después descifrarla mediante algún sistema criptoanalítico.
- Modificación del mensaje: capturar paquetes para luego ser borrados, manipulados, modificados o reordenados. En muchos sistemas criptográficos parte de la clave se envía en estos paquetes de forma cifrada. Solamente se tendrá que encontrar estos paquetes y comenzar a trabajar sobre ellos.
- Reactivación: Captura de paquetes y retransmisiones.
- Degradación: Técnicas para que el servicio se degrade.

El otro tipo de ataque criptoanalítico es el ataque activo, aquí el atacante no altera la comunicación solo la escucha o monitoriza, para obtener información. Estas técnicas son difíciles de detectar ya que no implican alteración de los datos. (Sánchez, J. M., 2011)

1.9 La criptografía y sus dos ramas.

La criptografía, encargada de proteger los mensajes que el criptoanálisis pretende conocer, se divide en dos ramas:

- Criptografía simétrica: Se refiere al conjunto de métodos que permiten tener una comunicación segura entre emisor y receptor, siempre y cuando con anticipación se hayan intercambiado la clave que es llamada clave simétrica. Es denominado cifrado simétrico, porque utiliza la misma clave para cifrar y descifrar un mensaje. Se conoce también como cifrado de clave privada. Se ha caracterizado por ser la más usada durante toda la historia, siendo implementada en diversos dispositivos manuales, mecánicos, eléctricos y algoritmos computacionales. La idea principal es aplicar diferentes funciones al mensaje que se quiere cifrar

de modo que solo se conozca una clave que se aplique de forma inversa para poder descifrar.

- Criptografía asimétrica: Este cifrado se caracteriza por utilizar dos claves diferentes para el emisor y receptor, entonces para cifrar mensajes se necesita una clave pública y para descifrar mensajes una clave privada. Nace con la finalidad de buscar métodos más prácticos para intercambiar claves simétricas. Aproximadamente en 1975, dos ingenieros de la Universidad de Stanford, Whitfield Diffie y Martin Hellman, publican un artículo llamado “New Directions in Cryptography” (Nuevas direcciones en Criptografía) que introdujo el concepto de criptografía de clave pública. Los algoritmos de cifrado con clave privada o mejor dicho los cifrados simétricos hasta el momento eran los únicos conocidos, pero ya no poseían las necesidades de seguridad para cifrado un mensaje.

Intercambio de clave de Diffie – Hellman:

1. Se seleccionan dos usuarios, A y B, un grupo finito cíclico, G, de orden n y un generador $\alpha \in G$.
2. A genera un número a, se calcula $\alpha^a \in G$ y se transmite a B.
3. B genera un número b, se calcula $\alpha^b \in G$ y se trasmite a A.
4. A recibe α^b y calcula $\alpha^{b^a} \in G$.
5. B recibe α^a y calcula $\alpha^{ab} \in G$.

Los dos usuarios A y B tienen un elemento en común y por propiedades de potencias se tiene $(\alpha^a)^b = (\alpha^b)^a = \alpha^{ab}$.

Conclusiones del capítulo:

Se sentaron las bases teóricas para la mejor comprensión de las temáticas a abordar.

Se demostró un novedoso algoritmo de factorización de vital importancia para cumplir el objetivo del trabajo.

Se trataron las ecuaciones diofánticas lineales y cuadráticas reducibles a Pell, mostrando soluciones obtenidas por simple inspección. Tomando en

consideración que ellas son la base de este algoritmo de factorización, se profundizará en la búsqueda de métodos de solución en próximos capítulos.

Se trató una función conocida como función de Euler, la cual devuelve la cantidad de factores coprimos al n insertado, función que genera el cuerpo en el que se trabajará en próximos capítulos.

Se trata el inverso modular basado en la noción de inverso usual, que será necesario en el desarrollo del trabajo.

Se evidencia la relación entre criptografía y criptoanálisis, siendo una la ciencia inversa a la otra. El trabajo, aunque pertenece al criptoanálisis, está basado en una de las dos ramas de la criptografía, la criptografía asimétrica. En especial, en uno de los algoritmos criptográficos más utilizados en la actualidad, algoritmo creado sobre los fundamentos teóricos de Diffie-Hellman.

Capítulo 2

Capitulo II: Construcción del algoritmo de ataque.

El método de Difiie- Hellman, aunque planteaba las bases teóricas para un nuevo tipo de criptografía, y aseguraba la dificultad para descifrar un mensaje cifrado así, no era un criptosistema, solo permite el intercambio de claves, no lleva a cabo cifrar ni descifrar.

2.1 Historia del RSA

Ronald Rivest, Adi Shamir and Leonard Adleman eran un equipo perfecto. Rivest era especialista en ciencias de la computación, muy ingenioso y creativo, tenía habilidad para aplicar nuevas ideas. Además estaba muy actualizado sobre los artículos que iban saliendo y creaba ideas en función de ellos. Shamir, también era especialista en ciencias de la computación, tenía un intelecto extraordinario, y una gran habilidad creando nuevas ideas y técnicas para resolver problemas. Él y Rivest querían llevar a la práctica las bases teóricas planteadas por Diffie-Hellman y para ello idearon una función que lo permitiría. Adleman era un matemático que poseía tres cualidades fundamentales: inteligencia, rigor y paciencia. Él puso gran empeño en eliminar los errores de la función ideada por Rivest y Shamir, y se aseguró de que no continuaran en la dirección equivocada. Rivest y Shamir pasaron un año creando su idea, y Adleman pasó otro año corrigiendo errores. Esto era un poco decepcionante, pero ellos sabían que cada falla que se iba presentando tenía una solución con fuertes bases matemáticas. En abril de 1977, Rivest, Shamir y Adleman, un poco decepcionados por no obtener ningún resultado, después de tanto tiempo, se plantearon si era posible encontrar la función de una sola dirección, que se pueda calcular el inverso solo si se tiene una información especial. Pero Rivest no se daba por vencido y esta cuestión no lo dejaba dormir, hasta que la nubosidad comenzó a aclarar y tuvo lo que algunos llamaron una revelación. Gastó el resto de esta noche tratando de formalizar su idea, y al amanecer ya había escrito su artículo completo. Aunque Rivest llegó a un resultado extraordinario, no hubiera sido posible sin la ayuda de Shamir y Adleman. El sistema criptográfico fue llamado RSA, por **R**ivest, **S**hamir y

Adleman. Este algoritmo estuvo patentado en los EEUU hasta el 2000, por lo que su uso estuvo restringido hasta esa fecha para el resto el mundo.

De entre todos los algoritmos asimétricos, quizás RSA sea el más sencillo de comprender e implementar.

2.2 ¿Cómo funciona RSA?

RSA tiene tres etapas entrelazadas entre sí: generar claves, cifrar el mensaje y descifrar el mensaje.

Antes de comenzar la primera etapa es necesario asignarle a cada letra del alfabeto un número, usualmente se le asocian números de dos cifras.

Para generar las claves cada usuario debe buscar dos números primos grandes y fuertes (es decir, con determinadas propiedades) que se denominarán p y q, luego se multiplicarán p y q, obteniendo $n = p * q$.

Se halla la función de Euler de n, $\phi(n) = \phi(p * q) = (p - 1) * (q - 1)$, por ser p y q números primos.

Luego se busca un número e, tal que $1 < e < \phi(n)$ y e sea coprimo con $\phi(n)$.

Por último, se halla d, tal que $d * e \equiv 1 mod(\phi(n))$, es decir d es el inverso módulo $\phi(n)$ de e.

La clave pública, que se publicará en un directorio, será (e, n) y la clave privada, que solo poseerá el usuario que generó las claves, será (d, n).

En la segunda etapa para cifrar, es necesario ya tener la lista de números D equivalentes al mensaje que se quiere cifrar, esta lista se divide en valores menores que n, $D = D_1 D_2 D_3 \dots D_k$, cada uno de estos valores D_i se le realiza la siguiente operación: ${D_i}^e mod(n) = C_i$, la lista de los C_i, es decir $C_1, C_2, C_3, \dots, C_k = C$ es el mensaje cifrado. Cada usuario que desee enviar un mensaje a un destinatario debe buscar la clave pública del mismo en el directorio, luego realizar este proceso y enviar el mensaje C obtenido al destinatario.

En la tercera etapa el destinatario y creador de ambas claves recibe el mensaje cifrado $C = C_1, C_2, C_3, \dots, C_k$ y realiza la siguiente operación ${C_i}^d mod(n) = D_i$, utilizando su clave privada, así completa el mensaje $D = D_1 D_2 D_3 \dots D_k$, solo faltará asignarle a cada número su letra correspondiente según la asignación antes realizada.

2.3 Envío de mensajes con RSA.

A continuación, la autora propone y desarrolla un ejemplo académico para ilustrar el funcionamiento de las etapas del RSA. Las denominaciones de Ana y Bob se corresponden con la terminología de A utilizada para el emisor y de B para el receptor.

Ana le quiere enviar el mensaje “HOLA BOB” a Bob. Ambos conocen el alfabeto que se utilizará para cifrar el mensaje.

Alfabeto:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
11	12	13	14	15	16	17	18	19	20	21	22	23	24

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
25	26	27	28	29	30	31	32	33	34	35	36	37	99

Al espacio se le asociará el 99.

Bob necesita crear su par de claves.

Primera etapa: Generación de claves.

Para menor complejidad se tomaran números primos pequeños.

Bob busca dos números primos $p = 13$ y $q = 19$, los multiplica, obteniendo $n = p * q = 13 * 19 = 247$

Halla la función de Euler de n, $\phi(247) = \phi(13 * 19) = (13 - 1) * (19 - 1) = 216$.

Luego Bob escoge un número $e = 7$, tal que $1 < e < \phi(n)$ y e sea coprimo con 216.

Por último, halla d tal que $d * 7 \equiv 1 mod(216)$, es decir d es el inverso módulo $\phi(n)$ de e, en este caso $d = 31$.

Bob publica $(7, 247)$ y guarda su clave privada $(31, 247)$.

Segunda etapa: Cifrado de mensaje.

Ana convierte su mensaje por el alfabeto conocido por ambos a una lista de números D que luego cifrará.

H	O	L	A		B	O	B
18	26	22	11	99	12	26	12

$$D = 1826221199122612$$

Ya teniendo D y la clave pública de Bob $(e, n) = (7, 247)$, Ana divide esta lista en valores $D_i < 247$, es decir, $D_1 = 182$, $D_2 = 62$, $D_3 = 211$, $D_4 = 99$, $D_5 = 122$, $D_6 = 61$ y $D_7 = 2$.

Realiza a cada D_i la siguiente operación: ${D_i}^7 mod(247) = C_i$

$$(182)^7 mod(247) = 182 = C_1$$

$$(62)^7 mod(247) = 244 = C_2$$

$$(211)^7 mod(247) = 185 = C_3$$

$$(99)^7 mod(247) = 44 = C_4$$

$$(122)^7 mod(247) = 8 = C_5$$

$$(61)^7 mod(247) = 139 = C_6$$

$$(2)^7 mod(247) = 128 = C_7$$

La lista de los C_i, es decir $C_1, C_2, C_3, \dots, C_k = C$ es el mensaje cifrado. Ana enviará a Bob $C = 182, 244, 185, 44, 8, 139, 128$.

Bob recibe el mensaje C de Ana; para leerlo debe descifrarlo primero utilizando su clave privada $(d, n) = (31, 247)$.

Tercera etapa: Descifrado de mensaje.

Bob realiza la siguiente operación ${C_i}^d mod(n) = D_i$,

$$(182)^{31} mod(247) = 182 = D_1$$

$$(244)^{31} mod(247) = 62 = D_2$$

$$(185)^{31} mod(247) = 211 = D_3$$

$$(44)^{31} mod(247) = 99 = D_4$$

$$(8)^{31} mod(247) = 122 = D_5$$

$$(139)^{31} mod(247) = 61 = D_6$$

$$(128)^{31} mod(247) = 2 = D_7$$

Ya obtenido $D = D_1 D_2 D_3 \dots D_k = 1826221199122612$, Bob debe dividir esta lista en números de dos cifras y asignarle a cada uno su letra correspondiente en el alfabeto.

18	26	22	11	99	12	26	12
H	O	L	A		B	O	B

Y así Bob lee el mensaje enviado por Ana.

2.4 Un algoritmo para atacar la seguridad del RSA.

Toda la seguridad del criptosistema de clave pública RSA se basa en la dificultad que posee la factorización de números grandes. Un algoritmo veloz para la factorización de enteros implicaría que este algoritmo criptográfico fuese inseguro.

Conocido el mensaje cifrado C y la clave pública del destinatario, si se quiere conocer la clave privada se debe seguir el siguiente algoritmo.

Algoritmo de ataque

Paso1: Factorizar n, es decir, se buscan p y q.

Paso2: Se calcula la función de Euler de n, $\phi(n)$.

Paso3: Se calcula d.

Paso4: Se calcula $C^d mod(n) = D$.

2.5 Factorización de n

Sea $n = p * q$ con p y q desconocidos.

De acuerdo con los resultados obtenidos en la demostración de la **Proposición 1.4.1,** para factorizar n se debe seguir el siguiente algoritmo:

Paso 1: Se llamará m a la parte entera de $\frac{n}{3}$**,** es decir, $m = \left\lfloor \frac{n}{3} \right\rfloor$

Paso 2: Si m par y si tiene solución la ecuación diofántica $3xy + x + y = m$

Entonces $p = 3x + 1$ y $q = 3y + 1$

Si no se resuelve la ecuación diofántica $3xy + 2x + 2y + 1 = m$ y $p = 3x + 2$ y $q = 3y + 2$

Si no se resuelve la ecuación diofántica $3xy + 2x + y = m$ y $p = 3x + 1$ y $q = 3y + 2$

Paso 3: Devolver p y q.

2.6 Ejemplos de factorización de n

Los tres ejemplos siguientes ilustran cada una de las alternativas contempladas en la demostración de la **Proposición 1.4.1.** En el primero de ellos, se comprueba la selección realizada en el ejemplo mostrado en el epígrafe 2.3.

Ejemplo 1:

Sea $n = 247$

Paso 1: $m = \left\lfloor \frac{247}{3} \right\rfloor = 82$

Paso 2: Como 82 es par, $3xy + x + y = 82$, $x = 4$ y $y = 6$ por tanto $p = 3 * 4 + 1 = 13$ y $q = 3 * 6 + 1 = 19$

Paso 3: 13 y 19.

Ejemplo 2:

Sea $n = 391$

Paso 1: $m = \left\lfloor \frac{391}{3} \right\rfloor = 130$

Paso 2: Como 130 es par, $3xy + x + y = 130$, no tiene solución en los naturales. Para $3xy + 2x + 2y + 1 = 130$, $x = 5$ y $y = 7$ por tanto $p = 3 * 5 + 2 = 17$ y $q = 3 * 7 + 2 = 23$

Paso 3: 17 y 23.

Ejemplo 3:

Sea $n = 221$

Paso 1: $m = \left\lfloor \frac{221}{3} \right\rfloor = 73$

Paso 2: Como 73 es impar, $3xy + 2x + y = 73$, $x = 4$ y $y = 5$ por tanto $p = 3 * 4 + 1 = 13$ y $q = 3 * 5 + 2 = 17$

Paso 3: 13 y 17.

2.7 Determinando d.

El otro problema para encontrar la clave privada d, es el cálculo de un inverso modular, es decir, encontrar d tal que $d * e \equiv 1 mod(\phi(n))$.

En este caso se resolverá el problema utilizando ecuaciones diofánticas lineales restringidas al conjunto de los números naturales. (Véase subepígrafe 1.6.1)

$d * e = k * \phi(n) + 1 \Leftrightarrow d * e - k * \phi(n) = 1$

Donde d y k son las variables.

Ejemplo:

Se analizará el ejemplo que se ha estado tratando durante todo el desarrollo del capítulo.

$e = 7$, $n = 247$, $p = 13$ y $q = 19$ (véase ejemplo 1 epígrafe 2.6)

$\phi(n) = (p - 1) * (q - 1) = 12 * 18 = 216$

$d * e - k * \phi(n) = 1 \Leftrightarrow d * 7 - k * 216 = 1$

Esta ecuación diofántica no posee una única solución, entre sus soluciones están:

$d = 31\ y\ k = 1$, $d = 67\ y\ k = 13$, $d = 103\ y\ k = 20$, $d = 139\ y\ k = 27$ y $d = 175\ y\ k = 34$

Pero, ¿cómo encontrar la clave privada d que posee el receptor para poder descifrar el mensaje?

Se analizará el caso obtenido en el ejemplo del epígrafe 2.3: $62 = D_2$

$$(62)^7 mod(247) = 244 = C_2$$

Realizando la operación de descifrado para cada uno de los d obtenidos en el **Ejemplo** de este epígrafe**.**

$$(244)^{31} mod(247) = 62$$

$$(244)^{67} mod(247) = 62$$

$$(244)^{103} mod(247) = 62$$

$$(244)^{139} mod(247) = 62$$

$$(244)^{175} mod(247) = 62$$

Evidentemente para cada uno de los d obtenidos la operación de descifrado devuelve el mismo resultado, por lo que se puede conjeturar que aunque la d no es única, cada una de ellas hace la misma función de clave privada.

A este conjunto de valores de d se conoce como Claves Privadas Parejas (CPP).

2.8 Claves privadas parejas.

Una clave privada pareja permite descifrar el cifrado C. El algoritmo RSA posee como mínimo una CPP. La existencia de estas CPP está dada en que e y d son inversos en el anillo generado por $\phi(n)$, mientras que el proceso de cifrado y descifrado se realiza en el anillo generado por n. Aunque la clave privada no cumple la unicidad, esta condición no compromete ni el efecto ni la seguridad del algoritmo, pues para conocer cualquier clave privada, sea la original o sea pareja, es necesario la factorización de n.

¿Cómo saber las CPP de una clave privada?

Se tienen e, n; se calcularán p y q.

Una vez obtenidos p y q, se realiza el siguiente algoritmo:

Algoritmo para encontrar CPP:

Paso 1: se calcula $\gamma = mcm[(p-1),(q-1)]$.

Paso 2: se calcula el inverso de e módulo γ, es decir, $d_\gamma * e \equiv 1 mod(\gamma)$.

Paso 3: se calcula la cantidad de CPP, $\lambda = \left\lfloor \frac{(n-d_\gamma)}{\gamma} \right\rfloor$

Paso 4: se calculan las CPP, $d_i = d_\gamma + i\gamma$, $1 < i < \lambda$.

Aplicación del algoritmo para encontrar CPP:

$e = 7$, $n = 247$, $p = 13$ y $q = 19$

Paso 1: $\gamma = mcm[(13-1),(19-1)] = mcm[12,18] = 36$

Paso 2: $d_{36} * 7 \equiv 1 mod(36)$, tomemos el valor $d_{36} = 31$.

Paso 3: $\lambda = \left\lfloor \frac{(247-31)}{36} \right\rfloor = 6$

Paso 4: $d_1 = 31 + 1 * 36 = 67$

$d_2 = 31 + 2 * 36 = 103$

$d_3 = 31 + 3 * 36 = 139$

$d_4 = 31 + 4 * 36 = 175$

$d_5 = 31 + 5 * 36 = 211$

$d_6 = 31 + 6 * 36 = 247$

2.9 Ejemplos de ataque a RSA.

Ana le quiere enviar un mensaje a Bob. Ambos conocen el alfabeto que se utilizará para cifrar el mensaje. Existe un intermediario desconocido que conoce el alfabeto, la clave pública, el mensaje cifrado y que desea conocer el mensaje original, que llamaremos ID.

Utilizando el ejemplo del epígrafe 2.3.

Sea $C = 182, 244, 185, 44{,}8, 139, 128$ el mensaje cifrado, $(7, 247)$ la clave pública y el alfabeto

A	B	C	D	E	F	G	H	I	J	K	L	M	N
11	12	13	14	15	16	17	18	19	20	21	22	23	24

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
25	26	27	28	29	30	31	32	33	34	35	36	37	99

Los pasos que sigue ID para obtener el mensaje original se describen en el siguiente algoritmo.

Aplicación de algoritmo de ataque

Paso 1: factoriza $n = 247$, es decir, busca p y q tales que $p * q = 247$, utilizando el algoritmo del epígrafe 2.5.

$p = 13$ y $q = 19$. (Véase el ejemplo 1 del epígrafe 2.6)

Paso 2: calcula la función de Euler de 247, $\phi(247) = 12 * 18 = 216$.

Paso 3: calcula d, donde $d * 7 = 1 mod\,(216)$, es decir resuelve la ecuación diofántica lineal $d * 7 + k * 216 = 1$, $d = 31$.

Paso 4: Una vez obtenida la clave privada, calcula $C^d mod(n) = D$ y obtiene el mensaje.

Conclusiones del capítulo

Se explican las tres etapas del RSA y se destacan sus relaciones.

Se propone y desarrolla un ejemplo académico para ilustrar el funcionamiento de las etapas del RSA. Este ejemplo cumple además el propósito de ejemplificar numéricamente los principales conceptos y métodos requeridos, de modo que ayudan a la comprensión de elementos teóricos con un elevado grado de abstracción, lo que le da valor metodológico al trabajo.

Se muestra la falta de unicidad de las claves privadas y se evidencia que esta condición no compromete ni el efecto ni la seguridad del algoritmo. Este hecho conduce a introducir la noción de claves privadas parejas.

Se aporta un algoritmo para determinar claves privadas parejas y se comprueba la coincidencia de los resultados que ofrece cuando se comparan con los valores que se obtienen al resolver la ecuación diofántica del ejemplo manejado.

Se propone, además, un algoritmo para atacar el criptosistema RSA, utilizando una novedosa forma para la factorización de enteros.

La forma de presentación conduce a percibir la necesidad del tratamiento computacional automatizado por la complejidad de los cálculos involucrados, aún en casos relativamente simples. En particular se observa que es necesaria la resolución de ecuaciones diofánticas lineales y cuadráticas reducibles a Pell, lo cual requiere la utilización de un software adecuado.

Capítulo 3

Capítulo III: Implementación computacional del algoritmo de ataque

Python es un lenguaje de programación de alto nivel que se puede usar para realizar todo tipo de tareas en múltiples plataformas. Su filosofía hace énfasis en que el código que se escriba sea lo más legible posible y su sintaxis permite expresar una idea en menos líneas de código que otros lenguajes como Java o C++. Es un lenguaje que está diseñado para ser ejecutado mediante un intérprete, en contraste con otros lenguajes compilados. Es multiparadigma, ya que soporta orientación a objetos, programación imperativa y programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

3.1 Software PYTHON: Características y facilidades.

Historia de Python

Python comenzó a concebirse a finales de los 80 como proyecto de código abierto y su implementación fue iniciada en diciembre de 1989 por Guido van Rossum en el CWI en Países Bajos, como un sucesor al lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba. Es administrado por la empresa Python software Foundation.

La comunidad Python ha llamado a Guido van Rossum el BDFL (benevolent dictator for life o dictador benevolente de por vida), debido a su rol central en la toma de decisiones del desarrollo de Python.

Python 2.0 fue lanzado el 16 de octubre de 2000, con grandes cambios y características como un recolector completo de basura y soporte para Unicode. Con el lanzamiento de esta versión, el proceso de desarrollo cambió y se volvió más transparente y más apoyado en la comunidad Python.

Python 3.0 (también llamado Python 3000 o py3k), fue lanzado -tras un largo período de prueba- el 3 de diciembre de 2008; se volvió incompatible hacia atrás.

Uso de Python

El lenguaje de programación Python es uno de los más usados en el mundo, según una medición de TIOBE Programming Community Index (2008) es el octavo más popular. Además es el tercero más popular en aquellos lenguajes que no basan su sintaxis gramatical en C.

Un estudio mostró que Python hace un uso de la memoria mejor que JAVA y no tan lejos de la eficiencia de C o C++.

Grandes organizaciones utilizan Python para algunos de sus productos como Google, Yahoo!, CERN, NASA, etc. También es utilizado en la computación científica gracias a librerías como NumPy, SciPy y Matplotlib. Es empleado en tareas de inteligencia artificial, como por ejemplo en tareas de procesamiento de lenguajes naturales.

Características de los lenguajes scripting

- Los scripts suelen escribirse más fácilmente, pero con un costo sobre su ejecución.
- Suelen implementarse con intérpretes en lugar de compiladores.
- Tienen fuerte comunicación con componentes escritos en otros lenguajes.
- Los scripts suelen ser almacenados como texto sin formato.
- Los códigos suelen ser más pequeños que el equivalente en un lenguaje de programación compilado.

Características generales de Python

- Lenguaje de programación de alto nivel del tipo scripting.
- Diseñado para ser fácil de leer y simple de implementar.
- Es código abierto (de libre uso).
- Puede ejecutarse en Mac, Windows y sistemas Unix; también ha sido portado a máquinas virtual JAVA y .NET.

- Es a menudo usado para desarrollar aplicaciones web y contenido web dinámico.
- Se utiliza para crear extensiones tipo plug-ins para programas de 2d y 3d como Autodesk Maya, GIMP, Blender, Inkscape, etc.
- Los scripts de Python tienen la extensión de archivo .PY, que pueden ser y ejecutados inmediatamente.
- Permite grabar programas compilados con extensión de archivo .PYC, los cuales suelen ser usados como módulo que pueden ser referenciados por otros programas Python.

Aplicaciones más comunes que utilizan Python:

- Aplicaciones: BitTorrent, Blender 3D, Calibre, Dropbox, MusicBrainz Picard, Ubuntu Software Center, YUM, etc.
- Aplicaciones web: GNU Mailman, OpenERP.
- Videojuegos: Civilization IV, Disney Toontown Online, Battlefield 2, Vega Strike.

Python es una opción magnífica para todo aquel que quiera iniciarse en la programación. Al contrario que en otros lenguajes, no se llena el código con símbolos, puntos y coma o corchetes, pues la sintaxis es muy limpia y resulta agradable de leer. Pero si hay algo que de verdad caracteriza a este lenguaje de programación, es su capacidad para ser utilizado en múltiples propósitos distintos. Para ello, se sirve de diferentes librerías que añadirán funciones adicionales a este lenguaje.

3.2 Opciones de Python para el diseño de la investigación.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones o bibliotecas entre ellas están NumPy y SymPy.

NumPy le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices. El ancestro de NumPy, Numeric, fue creado originalmente por Jim Hugunin con algunas contribuciones de otros desarrolladores. En 2005, Travis Oliphant creó NumPy incorporando características de Numarray en NumPy con algunas modificaciones. Usualmente se utiliza para cálculos científicos. Es determinante para el trabajo con listas.

Mientras que SymPy tiene como principal objetivo reunir todas las características de un sistema de álgebra computacional (CAS), ser fácilmente extensible y mantener el código todo lo simple que sea posible. SymPy no requiere ninguna biblioteca externa, salvo para soporte gráfico.

Sus capacidades básicas incluyen:

- El manejo de enteros de precisión arbitraria y de números racionales,
- La simplificación básica, expansión, sustitución básica,
- El manejo de funciones sobre el cuerpo de los complejos,
- La derivación y expansión en series de Taylor o de Laurent,
- Los símbolos no conmutativos.

Dentro de sus módulos incluye:

- Funciones como factorial, zeta, Legendre, entre otras;
- límites,
- integración,
- divisibilidad y factorización de polinomios,
- resolución de ecuaciones diofánticas, algebraicas, diferenciales y sistemas,
- operaciones con matrices simbólicas,
- Álgebra de Dirac y de Pauli,
- Representación gráfica (en 2D y en 3D).

Para la resolución de las ecuaciones cuadráticas reducibles a Pell se utiliza la función $diophantine$ importada de la biblioteca $sympy$, función que devuelve soluciones enteras, por lo que la autora crea una función, int_solve, que

restringe estas soluciones al conjunto de los naturales, hipótesis necesaria en el nuevo algoritmo de factorización.

```
In [3]: diophantine(3*x*y+x+y-5)
Out[3]: {(-3, -1), (-1, -3), (0, 5), (1, 1), (5, 0)}
```

```
In [4]: int_solve(3*x*y+x+y,5)
Out[4]: (1, 1)
```

Para la resolución de ecuaciones diofánticas lineales para el cálculo de la clave privada se utiliza la función $diop\_lineal$ de la misma biblioteca.

```
In [5]: diop_linear(x*7+y*216-1)
Out[5]: (216*t_0 + 31, -7*t_0 - 1)
```

Esta función devuelve la forma general que deben tener las soluciones.

3.3 Implementación del algoritmo sobre Python.

Este algoritmo está integrado por dos partes fundamentales. La primera parte llamada $melb\_1$ utiliza como variables los dos elementos de la clave pública e y n.

Primero factoriza n utilizando la función $integer\_factor$ propuesta por la autora en la que se utiliza la nueva forma de factorización basada en ecuaciones diofánticas cuadráticas reducibles a Pell.

```
def integer_factor(n):
    m=n//3
    if m%2:
        s=int_solve(f_(2),m)
        p=3*s[0]+1
        q=3*s[1]+2
        return p,q
    else:
        k=int_solve(f_(0),m)
        if k==None:
            s=int_solve(f_(1),m)
            p=3*s[0]+2
            q=3*s[1]+2
            return p,q
        else:
            p=3*k[0]+1
            q=3*k[1]+1
            return p,q
```

Luego calcula d, resolviendo una ecuación diofántica lineal.

```
def melb_1(e,n):
    s=integer_factor(n)
    phi=(s[0]-1)*(s[1]-1)
    set_sol = diop_linear(x*e+y*phi-1)
    print(set_sol)
```

Devolviendo el valor de d en función de un parámetro t_0 y con la variación de este t_0 el algoritmo permite encontrar las claves privadas parejas.

Ejemplo de corrida de $melb\_1$:

```
In [6]: n=247

In [7]: e=7

In [8]: melb_1(e,n)
(216*t_0 + 31, -7*t_0 - 1)

In [9]: d=216*t_0 + 31 para t_0=0 d=31
```

Luego se define la segunda parte del algoritmo $melb\_2$, función que tiene como variables el d obtenido anteriormente, n de la clave pública y C el mensaje cifrado del cual se desea conocer el texto en claro.

```
def melb_2(n,d,c):
    D=[modpow(i,d,n) for i in c]
    print(D)
```

Para lograr el propósito fue necesaria la implementación de otras dos funciones, ya que C no es un valor sino una lista de valores a los que se le debe realizar la misma operación.

La primera fue $modpow$ que tiene como objetivo realizar la potencia modulo n.

```
def modpow(base, exp, mod):
    if mod==1:
        return 0
    else:
        r = 1
        base = base % mod
        while exp > 0:
            if exp % 2 == 1:
                r = (r * base) % mod
            exp = exp >> 1
            base = (base**2) % mod
        return r
```

La segunda llamada $nmodpow$ ejecuta la función $modpow$ para cada elemento de la lista C.

```
def nmodpow(bases, exp, mod):
    # wrapper de la funcion modpow:
    return [modpow(base, exp, mod) for base in bases]
```

Ejemplo de corrida de $melb\_2$:

```
In [10]: d=31

In [11]: n=247

In [12]: c=[182,244,185,44,8,139,128]

In [13]: melb_2(n,d,c)
[182, 62, 211, 99, 122, 61, 2]
```

Solo quedaría transformar la lista de números al texto en claro utilizando el alfabeto.

3.4 Ejemplos de aplicación del algoritmo de ataque.

Si se observan las ilustraciones analizadas en el epígrafe anterior se podrá constatar que coinciden con el ejemplo analizado en el **capítulo II.** El siguiente ejemplo muestra el desarrollo del citado ejemplo paso a paso, desde el punto de vista de su implementación computacional.

Ejemplo 3.4.1

El intermediario tiene como datos el alfabeto:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
11	12	13	14	15	16	17	18	19	20	21	22	23	24

Ñ	O	P	Q	R	S	T	U	V	W	X	Y	Z	
25	26	27	28	29	30	31	32	33	34	35	36	37	99

La clave pública $(e,n) = (7,247)$ y el mensaje cifrado $C = 182, 244, 185, 44,8, 139, 128$.

Primero debe encontrar los factores de $n = 247$ para calcular $\phi(n)$. Operación que realiza utilizando la función $integer\_factor$.

```
In [14]: integer_factor(247)
Out[14]: (19, 13)
```

Para calcular $\phi(n)$ utiliza la función $melb\_1$ donde además de calcular $\phi(n)$, calcula la forma general de la clave privada. Luego dando valores a t_0 se obtiene un valor específico de d.

```
In [25]: melb_1(7,247)
(216*t_0 + 31, -7*t_0 - 1)

In [26]: phi=216

In [27]: d=216*t_0 + 31 para t_0=0 d=31
```

Luego utilizando la función $melb\_2$ se calcula ${C_i}^d mod(n)$ para cada elemento de la lista C.

```
In [29]: c=[182,244,185,44,8,139,128]

In [30]: melb_2(247,31,c)
[182, 62, 211, 99, 122, 61, 2]
```

Obteniendo la lista $D = 182, 62, 211, 99, 122, 61, 2$, la que se debe dividir en valores de dos cifras por ser ese el tamaño de los valores asignados en el alfabeto.

$$D = 18, 26, 22, 11, 99, 12, 26, 12 = ''hola\ bob''$$

Se puede verificar que el mensaje que devuelve el algoritmo coincide totalmente con el mensaje original. (Véase **epígrafe 2.3**).

Sea el alfabeto siguiente común para todos los ejemplos restantes:

a	b	c	d	e	f	g	h	i
33205	33206	33207	33208	33209	33210	33211	33212	33213

j	k	l	m	n	o	p	q	r
33214	33215	33216	33217	33218	33219	33220	33221	33222

s	t	u	v	w	x	y	z	espacio
33223	33224	33225	33226	33227	33228	33229	33230	33299

Ejemplo 3.4.2:

Este ejemplo muestra un n que supera en trece cifras al n del ejemplo anterior, por lo que se pretende valorar el tamaño de n en la implementación del algoritmo de factorización.

Sea $e = 1223$ y $n = 2791857104398003$ la clave pública y sea

$c = 2551533115133743{,}840218092820863{,}1150487821389207$ el mensaje cifrado. Se desea encontrar el texto en claro.

Primero se ejecuta la factorización de $n = 2791857104398003$ para calcular $\phi(n)$, mediante la operación realizada utilizando la función $integer\_factor$, obteniendo como resultado $p = 86028121$ y $q = 32452843$.

Para calcular $\phi(n)$ la autora utiliza la función $melb\_1$ donde además de calcular $\phi(n)$, se calcula la forma general de la clave privada. Luego dando valores a t_0 se obtiene un valor específico de d. $d = 2791856985917040 * t\_0 - 579829660198633$, para $t\_0 = 1$ se obtiene $d = 2212027325718407$.

Luego utilizando la función $melb\_2$ se calcula $C_i{}^d mod(n)$ para cada elemento de la lista C.

El programa devuelve $D = 332293320533217332133321 6$, se debe dividir D en valores de cinco cifras, por ser ese el tamaño de los valores asignados a cada letra en el alfabeto y luego buscar dichos valores en el alfabeto.

$D_1 = 33229$, $D_2 = 33205$, $D_3 = 33217$, $D_4 = 33213$, $D_5 = 33216$

$D_1 = y,\ D_2 = a,\ D_3 = m,\ D_4 = i,\ D_5 = l$, el texto en claro es "$yamil$".

Ejemplo 3.4.3:

Sea $e = 3571$ y $n = 147453043535608653071$ la clave pública y sea

$C = 263709405380930833609445548550451741414115600970419916762399264949751146865820 78$ el mensaje cifrado y se desea encontrar el texto en claro.

El programa devuelve $D = 33209332233329933209\ 33216332993322033222, 33213332173320933222, 33299332083321333205$, se debe dividir D en valores de cinco cifras y luego buscar dichos valores en el alfabeto.

$D_1 = 33209,\ D_2 = 33223,\ D_3 = 33229,\ D_4 = 33209,\ D_5 = 33216,\ D_6 = 33299,$
$D_7 = 33220, D_8 = 33222, D_9 = 33213, D_{10} = 33217, D_{11} = 33209, D_{12} = 33222$

$D_{13} = 33299, D_{14} = 33208, D_{15} = 33213, D_{16} = 33205.$

e	s	espacio	e	l	espacio	p	r
33209	33223	33299	33209	33216	33299	33220	33222

i	m	e	r	espacio	d	i	a
33213	33217	33209	33222	33299	33208	33213	33205

El texto en claro es "$es\ el\ primer\ dia$".

Ejemplo 3.3.4:

Sea $e = 3571$ y $n = 258818587669079262174571213$ la clave pública y sea

$C =$
$78604960306676055743078800174990326725008755577366072243046585893727951076103063493082446159880305447962 6$
el mensaje cifrado y se desea encontrar el texto en claro.

El programa devuelve $D = 33209332233329933209\ 33216332993322033222\ 33213332173320933222\ 33299332083321333205$, se debe dividir D en valores de cinco cifras y luego buscar dichos valores en el alfabeto.

$D_1 = 33209,\ D_2 = 33223,\ D_3 = 33229,\ D_4 = 33209,\ D_5 = 33216,\ D_6 = 33299,$
$D_7 = 33220, D_8 = 33222, D_9 = 33213, D_{10} = 33217, D_{11} = 33209, D_{12} = 33222$

$D_{13} = 33299, D_{14} = 33208, D_{15} = 33213, D_{16} = 33205.$

e	s	espacio	e	l	espacio	p	r
33209	33223	33299	33209	33216	33299	33220	33222

i	m	e	r	espacio	d	i	a
33213	33217	33209	33222	33299	33208	33213	33205

El texto en claro es "$es\ el\ primer\ dia$".

Se puede observar que en los ejemplos 3.4.3 y 3.4.4 a pesar de que el texto en claro coincide hay una gran diferencia en cuanto a los textos cifrados y esto se debe a la diferencia que existe en la clave pública n escogida en cada uno de los casos. Se evidencia además que esta condición tan importante para la seguridad del algoritmo RSA no dificulta la efectividad del algoritmo propuesto.

3.5 Análisis de resultados.

Como se observa en los ejemplos anteriores el algoritmo permite dado un mensaje cifrado, un alfabeto y la clave pública del destinatario conocer el mensaje descifrado. En este algoritmo juega un papel fundamental la factorización de enteros, por lo que además se implementó un programa que utilizando la teoría del **capítulo I** sobre la nueva factorización de enteros y siguiendo el algoritmo presentado en el epígrafe 2.5 del capítulo II permite encontrar los factores primos de n y así calcular la función $\phi(n)$ que será indispensable para la determinación de la clave privada con la que luego se descifrará el mensaje. En cada ejemplo se muestra la eficacia de este algoritmo de factorización y esto se demuestra cuando la función $melb\_1$ devuelve correctamente la clave privada.

Durante la ejecución del algoritmo se utilizó el Wolfram Mathematica 11 (mediante el acceso a la nube libre Wolfram Alpha), para generar los números primos que serían multiplicados para obtener el n que se le entraría al

algoritmo propuesto. Para ello se utilizó la función $Prime$ que se le introduce un entero k y devuelve el primo que ocupa el lugar k dentro de la sucesión de primos. El Wolfram Mathematica 11 generó primos hasta $k = 6 * 10^{11}$, resultado mostrado en el **ejemplo 3.3.4.**

Conclusiones del capítulo

En la práctica el algoritmo ha cumplido con todas las condiciones teóricas propuestas por la autora, resolviendo inicialmente las ecuaciones diofánticas cuadráticas reducibles a Pell que permiten la factorización, utilizando luego estos factores para el cálculo de la función Phi de Euler, resultado que se necesitará para la resolución de una ecuación diofántica lineal cuyas incógnitas son la clave privada d y un valor k, que no se utilizará en el algoritmo. Después de obtener el algoritmo procede a realizar la operación de descifrado del mensaje C, mediante inversión modular tal y como se proponía en la teoría, realizando estas operaciones con éxito.

Conclusiones y recomendaciones

Conclusiones:

Se demostró un novedoso algoritmo de factorización en el cual las ecuaciones diofánticas cuadráticas reducibles a Pell juegan un papel primordial.

Se trataron las ecuaciones diofánticas lineales y cuadráticas reducibles a Pell, y se ilustraron las soluciones obtenidas por simple inspección. Tomando en consideración que ellas son la base del algoritmo de factorización, sus soluciones se obtuvieron en ejemplos posteriores a partir del software libre Python.

Se muestra la falta de unicidad de las claves privadas y se evidencia que esta condición no compromete ni el efecto ni la seguridad del algoritmo. Este hecho condujo a introducir la noción de claves privadas parejas.

Se aportó un algoritmo para determinar claves privadas parejas y se comprobó la coincidencia de los resultados obtenidos cuando se comparan con los valores determinados al resolver la ecuación diofántica del ejemplo manejado.

Se propone, además, un algoritmo para atacar el criptosistema RSA, utilizando esta novedosa forma para la factorización de enteros

En la práctica el algoritmo cumplió con todas las condiciones teóricas propuestas por la autora, resolviendo inicialmente las ecuaciones diofánticas cuadráticas reducibles a Pell que permiten la factorización, utilizando luego estos factores para el cálculo de la función Phi de Euler, resultado que se necesitará para la resolución de una ecuación diofántica lineal cuyas incógnitas son la clave privada d y un valor k, que no se utilizará en el algoritmo. Después de obtener el algoritmo procede a realizar la operación de descifrado del mensaje C, mediante inversión modular tal y como se proponía en la teoría, realizando estas operaciones con éxito.

Recomendaciones:

Implementar una nueva función en Python para la resolución más eficiente de las ecuaciones diofánticas, específicamente las reducibles a Pell.

Fortalecer las restricciones para la determinación de la clave pública con el objetivo de adicionar seguridad al RSA.

Bibliografía

Bibliografía:

A. Menezes, P. v. O. a. S. V. (1997). Handbook of Applied Cryptography, CRC Press.

Acosta, E. S. (2012). Criptoanálisis más utilizados en la actualidad. Criptografía y teoría de códigos.

Angel, J. d. J. A. (2008). Criptografía para principiantes.

Ayuso, A. (2009). Una simulación simplificada de la criptografía.

Bahit, E. (2012). Curso: Python para principiantes, SafeCreative.

Benjamin Baumslag, B. C. (2009). Theory and problems of Group Theory, Schaum's Outline Series.

García, R. d. M. (2008). Criptografía Clásica y Moderna, Septem Ediciones.

Garret, P. (1998). Intro Abstract Algebra.

Hector Corrales Sánchez, C. C. R., Alejandro Cuevas Notario (2006). "Criptografía y Métodos de Cifrado."

I. N. Bronshtein, K. A. S., G.Musiol, H.Muehling (2005). Handbook of Mathematics, Spinger.

López, M. J. L. (2001). Criptografía y seguridad en computadoras.

Mao, W. (2003). Modern Cryptography: Theory and Practice, Prentice Hall PTR.

Martín, C. A. R. S. (2016). Introducción a la criptografía. Departamento de Ciencias de la Educación, Universidad del Bio-Bio. Profesor de escuela media

Nápoli, P. d. (2014). Una introducción Matemática la Criptografía.

Olivos, V. C. (2005). Manual de Criptografía. S. d. Perú, Dirección Nacional de Programa de Jóvenes.

Pardo, T. J. R. (2008) Introducción a Python.

Rodó, D. M. (2017). El lenguaje Python, Universitat Oberta de de Catalunya.

Rossum, G. v. (2009). El tutorial de Python.

Sánchez, J. M. (2011). "Historias de matemáticas. Riemann y los números primos." Pensamiento Matemático.

Schneider, B. (1996). Applied Cryptography. Protocols, Algorithms, and Source, John Wiley & Sons, Inc.

Sergio Paque Martín, D. A. (2009). Programación Declarativa Avanzada.

Titu Andreescu, D. A., Ion Cucurezeanu (2010). An introduction to Diophantine Equations, Birkhäuser. Springer.

Vera Delgado, R. P. (2006) Introducción a la criptografía: tipos de algoritmos.

Yeste, N. V. (2011). Propiedad de los números primos de Fermat.