Universidad Autónoma Metropolitana Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Computación

Diseño e implementación de los algoritmos de encriptación RSA y Hill.

Modalidad: Proyecto Tecnológico
Trimestre 2017-Invierno

Alumno
David Rosales Serrano
2112001995

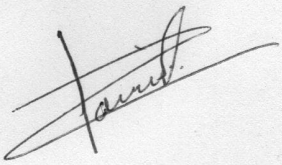
Asesor
Rogelio Herrera Aguirre

Abril de 2017

México CDMX

Yo,ROGELIO HERRERA AGUIRRE, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, DAVID ROSALES SERRANO, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Dedicado a mis padres
David y Leticia, quienes a lo largo de estos 24 años
me han brindado todo su amor, apoyo, confianza y
comprensión,

A mis hermanos Fernando y Jessica por su motivación y amor,
al igual que a toda mi familia,

A mi mejor amiga y novia Karla Hernández quien desde hace 8 años
me a brindado todo su amor y apoyo,

A mi asesor por su entrega y participación para
la realización de este proyecto,

Y a Dios por permitirme estar rodeado de la gente
que amo y dejarme lograr mis sueños.

Resumen

Desde tiempos remotos uno de los principales problemas a resolver ha sido la seguridad de la información, ya que se tiene documentado que por cuestiones militares, religiosas y comerciales se impulsó el uso de escrituras secretas. El primer caso de uso de encriptación se remonta durante la guerra entre Atenas y Esparta, donde se usó la escilata, que era un palo en el cual se le enrollaba en espiral una tira de cuero, donde se escribía el mensaje en forma paralela al eje del palo y este solo podía descifrarse si el receptor tenía un palo con las mismas características que el emisor. [8]

Es por eso, que hoy en día, para poder proteger cualquier tipo de información es necesario encriptarla, usando las técnicas modernas de encriptamiento, ya sea algoritmos asimétricos de llave pública, o bien algoritmos simétricos de llave privada, cada uno de los cuales presenta diferentes ventajas e inconvenientes según la aplicación a realizar.

En cualquiera de los casos antes anotados es de fundamental importancia el uso, y comprensión, de la aritmética de los campos finitos, aritmética modular.

Tomando en cuenta los conceptos anteriores, y considerando que en nuestros cursos de la carrera de Ingeniería en Computación no se contemplan los temas de aritmética modular, lo que se busca en este proyecto es estudiar y presentar dicha aritmética aplicada a dos algoritmos emblemáticos: el RSA de llave pública y el Hill de llave privada, e implementarlos, describiendo las ventajas y cualidades de ambos.

Índice general

Capítulo 1

Introducción

La palabra criptografía proviene del griego Kryptos (ocultar) y grafos (escribir), que literalmente, significa escritura oculta, por ello, podríamos decir que la criptografía, es la ciencia que cifra y descifra información, utilizando técnicas matemáticas que hacen posible el intercambio de mensajes, de tal manera, que sólo puedan ser leídos por las personas a las que va dirigido dicho mensaje. [1]

La criptografía, se puede clasificar históricamente en dos: La criptografía clásica y la criptografía moderna. La clásica, es aquella que se utilizó hasta la mitad del siglo XX, y también es conocida como criptografía no computarizada. Por otro lado, la criptografía moderna se inició después de los siguientes tres hechos: la publicación de la "Teoría de la Información"por Shannon, la aparición del estándar del sistema de cifrado DES (Data Encryption Standard), en 1974, y por último el estudio realizado por Whitfield Diffie y Martin Hellman sobre la aplicación de funciones matemáticas de un solo sentido, a un modelo de cifrado, denominado cifrado de llave pública en 1976. [8]

La criptografía moderna se clasifica en dos grupos: la criptografía simétrica y la asimétrica, la primera de éstas se caracteriza por un criptosistema en el cual existe una única llave secreta que deben compartir el emisor y el receptor, por lo que en este caso un problema fundamental para la seguridad consiste en mantener dicha llave en secreto, mientras que la criptografía asimétrica se distingue por ser un criptosistema donde el usuario crea un par de llaves, una privada y una pública, usándolas para descifrar y encriptar, respectivamente, los mensajes a comunicar; en este caso se minimiza el problema de la secrecía de la llave privada.

Una relación de gran utilidad para el desarrollo de las técnicas criptográficas es la relación de congruencia módulo m, definida dentro del anillo de los enteros, $\mathbb{Z}$, como sigue:

$$a \equiv b \,(mod\, m) \iff m|(b-a)$$

Esta relación resulta ser de equivalencia, y al conjunto de clases de equivalencia, *conjunto cociente*, $\mathcal{A}_m$, se le puede dotar de las operaciones suma y producto, inducidas por las operaciones respectivas de $\mathbb{Z}$, con tales operaciones $\mathcal{A}_m$ resulta ser un anillo, en el caso de que el módulo sea un número primo p, entonces la estructura definida por las operaciones resulta ser de campo y se obtiene el campo $\mathbb{F}_p$ con p elementos, debido a que

sobre estos campos, para cualquier entero positivo n, se pueden construir polinomios irreducibles de grado n, entonces se pueden construir campos finitos, $\mathbb{F}_{p^n}$, con p^n elementos, para cualesquiera p primo y n entero no negativo, la aritmética modular de este tipo de campos, en particular la relacionada con el problema del logaritmo discreto, así como la aritmética de los grupos definidos por las curvas elípticas sobre estos campos, han sido utilizadas en una variedad de algoritmos criptográficos, entre ellos el AES, el IDEA y el RSA. [13]

Por otro lado la aplicación del Álgebra Lineal de los espacios vectoriales definidos sobre los campos finitos, dio lugar a una técnica de encriptamiento, definida por Lester S. Hill en 1929, [12], [13], la cual es fundamento de diversos algoritmos de encriptamiento simétricos.

En el presente proyecto, se implementarán dos algoritmos criptográficos, el RSA y el Hill, se presentará también una descripción de sus respectivas cualidades, realizando una comparación entre éstas. Herramientas fundamentales, en la definición e implementación de los algoritmos anotados, son la aritmética modular y el Álgebra Lineal sobre campos finitos, razón por la cual en este proyecto se presentan los elementos necesarios de tales herramientas.

1.1. Antecedentes.

La Criptografía nace debido a que el hombre a lo largo del tiempo se ha visto en la necesidad de comunicar información confidencial a otros individuos ya sea por motivos militares, diplomáticos, comerciales, etc., en donde mantener la información en secreto es la pauta para conservar la integridad de un individuo o en ocasiones de una comunidad completa.
Es por eso que para mantener en secreto la información se emplean sistemas criptográficos, como lo son, los sistemas simétricos de llave privada como el RSA y los asimétricos de llave pública como lo es el algoritmo Hill.

1.2. Justificación.

El objetivo de este proyecto tuvo como finalidad el programar dos algoritmos criptográficos, el RSA y el Hill, el primero usa llave privada mientras el segundo usa llave pública. Los algoritmos desarrollados fueron probados varias veces para ver las cualidades de cada uno y así desarrollar la tabla de ambos algoritmos, con sus cualidades de cada uno.

1.3. Objetivos.

Objetivo General:

La realización de programas que implementen el algoritmo asimétrico RSA y el algoritmo simétrico Hill.

Objetivos específicos:

- Describir el marco teórico de la aritmética modular para los algoritmos RSA y Hill.
- Diseñar e implementar un módulo para convertir un texto a caracteres numéricos.
- Diseñar e implementar un módulo para realizar la aritmética modular.
- Diseñar e implementar un módulo para cifrar los números usando los algoritmos RSA y Hill.
- Diseñar e implementar un módulo para descifrar los números usando los algoritmos RSA y Hill.
- Diseñar e implementar un módulo para convertir caracteres numéricos a texto.
- Describir y comparar las cualidades de cada uno de los algoritmos implementados.

Capítulo 2

Aritmética entera y modular.

2.1. El anillo de enteros $\mathbb{Z}$.

El conocido conjunto de los números enteros, denotado usualmente $\mathbb{Z}$, junto con las operaciones aritméticas básicas, a saber suma y producto, poseé una rica estructura algebraica, para iniciar se dice que la terna $(\mathbb{Z}, +, \cdot)$, es un anillo conmutativo con unidad, ya que cumple el siguiente conjunto de propiedades.

Cerradura de la suma.	Cerradura del producto.
$(\forall a, b \in \mathbb{Z})(a + b \in \mathbb{Z})$	$(\forall a, b \in \mathbb{Z})(a \cdot b \in \mathbb{Z})$
Conmutatividad de la suma.	Conmutatividad del producto.
$(\forall a, b \in \mathbb{Z})(a + b = b + a)$	$(\forall a, b \in \mathbb{Z})(a \cdot b = b \cdot a)$
Asociatividad de la suma.	Asociatatividad del producto.
$(\forall a, b, c \in \mathbb{Z})([a + b] + c = a + [b + c])$	$(\forall a, b, c \in \mathbb{Z})([a \cdot b] \cdot c = a \cdot [b \cdot c])$
Existencia del neutro aditivo.	Existencia del neutro multiplicativo.
$(\exists 0 \in \mathbb{Z})(\forall a \in \mathbb{Z})(a + 0 = a)$	$(\exists 1 \in \mathbb{Z})(\forall a \in \mathbb{Z})(a \cdot 1 = a)$

Existencia de inversos aditivos.
$(\forall a \in \mathbb{Z})(\exists (-a) \in \mathbb{Z})([a + (-a)] = 0)$
Distributividad de la suma respecto del producto.
$(\forall a, b, c \in \mathbb{Z})([a + b] \cdot c = a \cdot c + b \cdot c)$

De $\mathbb{Z}$, se dice que es un anillo conmutativo con unidad, dado que el producto es conmutativo y que existe el neutro multiplicativo, si una estructura $(A, +, \cdot)$ donde $+$ y $\cdot$ son operaciones definidas sobre el conjunto A, tales que se cumplen todas las propiedades antes enlistadas, salvo la existencia de neutro multiplicativo, entonces, se dice anillo conmutativo, si en cambio, sólo no cumpliera la conmutatividad del producto, se conoce como anillo con unidad, finalmente si cumple las propiedades del listado, salvo la existencia de neutro multiplicativo y la conmutatividad del producto, se dice simplemente anillo.

Una propiedad de gran importancia del anillo de enteros $\mathbb{Z}$, es la siguiente:

$$(\forall a, b \in \mathbb{Z})(a \cdot b = 0 \iff a = 0 \,\text{o}\, b = 0)$$

Cuando un anillo conmutativo con unidad cumple tal propiedad, se dice de él que es un Dominio Entero, como veremos en la sección (2.2), existen anillos conmutativos con unidad que no son dominios enteros.

Por otro lado cuando un anillo a, ademas de las diez propiedades listadas al inicio de esta sección, cumple con la existencia de inversos multiplicativos, como a continuación se indica, se dice que es un Campo.

$$(\forall a \in A)(a \neq 0 \Rightarrow (\exists b \in A)(a \cdot b = 1))$$

Los campos mas conocidos son: el campo de los números racionales $\mathbb{Q}$, el campo de los números reales $\mathbb{R}$ y el de los números complejos $\mathbb{C}$, todos estos son camos infinitos, en la criptografía, son de particular interés los campos finitos, estos son eintroducidos en la sección (2.3).

Un par de conceptos de suma importncia en la aritmética de los enteros, son el de divisibilidad y el de primalidad. los cuales se definen a continuación.

Definición 2.1.1. Dados $a, b \in \mathbb{Z}$, decimos que a divide a b, lo cual se denota $a|b$, si existe $q \in \mathbb{Z}$, tal que, $b = qa$. ■

Observación 2.1.1.

- Paro todo entero a, $1|a$, $a|a$ y $a|0$.
- Si $a|0$, entonces, $a = 0$.
- Si $a|b$ y $a|c$, entonces, $a|(b + c)$.
- Si $a|(b + c)$ y $a|c$, entonces, $a|b$.

■

Definición 2.1.2. Un entero p, se dice primo si es mayor que 1, y sus únicos divisores positivos son 1 y p mismo. ■

A continuación se enuncia un resultado conocido como el *Teorema Fundamental de la Aritmética*.

Teorema 2.1.1. Para todo entero m, diferente de 0, 1 y -1, existe una factorización única en primos, como se indica a continuación

$$m = \pm 1 p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$$

donde los p_i son primos diferentes dos a dos, y los exponentes e_i enteros positivos, la unicidad de esta factorización es salvo el orden de los facores. ■

Otro resultado básico de la aritmética entera, es el conocido algoritmo de la división, connocido como *Algoritmo de Euclides*.

Algoritmo 2.1.1. Para a y b enteros, donde $b \neq 0$, entonces existen q y r enteros tales que

$$a = qb + r, \text{ con } 0 \leq r < |b|$$

A continuación se muestra un algoritmo que calcula q y r, para el caso en que a y b no son negativos, los otros casos son variantes de éste.

1. Entrada a, b, con $a \geq 0$ $b > 0$.
2. $r \leftarrow a$, $q \leftarrow 0$
3. Mientras $r \geq 0$ haga:
   a) $q \leftarrow q + 1$
   b) $r \leftarrow r - b$
4. $q \leftarrow q - 1$
5. $r \leftarrow r - b$ ■

Un concepto de gran utilidad dentro de la aritmética modular, es el de máximo común divisor, a continuación se presenta, junto con algunos resultados que serán usados en las secciones (2.2) y (2.3).

Definición 2.1.3. Para cualesquiera par de enteros a, b, definimos el conjunto de divisores comunes como:

$$D_c(a,b) = \{c \in \mathbb{Z} \,:\, c|a \,\text{y}\, c|b\}$$

■

Se puede ver que $D_c(0,0) = \mathbb{Z}$, en cualquier otro caso $D_c(a,b)$ es acotado y se cumple que para todo $d \in D_c(a,b)$, $|d| \leq máx\{|a|,|b|\}$,por otro lado también se cumple $D_c(a,b) = D_c(|a|,|b|)$, de la propiedad de acotamiento del conjunto de divisores comunes, tiene sentido la siguiente definición.

Definición 2.1.4. Dados a y b,enteros arbitrarios definimos el máximo común divisor, como sigue:

$$mcd(a,b) = \begin{cases} máx D_c(a,b) & \text{si } a \neq 0 \text{ o } b \neq 0 \\ 0 & \text{si } a = b = 0 \end{cases}$$

■

A continuación se presentan algunas propiedades del máximo común divisor.

Lema 2.1.1. Dados a y b enteros arbitrarios, y $d = mcd(a,b)$, se cumplen:

1. Si $u \in D_c(a,b)$, entonces, $d|u$.
2. Si $a = 0$, entonces, $d = |b|$.
3. $d = mcd(|a|,|b|)$
4. Existen enteros α y β, tales que: $d = \alpha a + \beta b$. ■

De hecho el máximo común divisor, también puede caracterizarce, mediante, el sigiente resultado.

Lema 2.1.2. Dados a y b enteros arbitrarios, se tiene:

$$d = mcd(a,b) \iff d = mín\{\omega \in \mathbb{N} \,:\, \omega = u \cdot a + v \cdot b, \text{ para, } u,v \in \mathbb{Z}\}$$

Del lema anterior, se puede observar la razón de la definición de máximo común divisor (2.1.4), en el caso en que tanto a como b son iguales a cero, a continuación se presenta un algoritmo que permite calcular el máximo común divisor de dos enteros positivos, así como de los coeficinetes que lo expresan como combinación de tales enteros.

Algoritmo 2.1.2. Dados a y b enteros positivos, se obtienen d, α y β, que cumplen:

$$d = mcd(a, b) = \alpha a + \beta b$$

1. $r \leftarrow a,\ r' \leftarrow b.$
2. $s \leftarrow 1,\ \ t \leftarrow 0$
3. $s' \leftarrow 0,\ \ t' \leftarrow 1$
4. Mientras $r' \neq 0$ haga
5. $\{(q, r'') \leftarrow CocRes(r, r'),\ \ (r, s, t, r', s', t') \leftarrow (r', s', t', r'', s - s'q, t - t'q)\}$
6. $(d, \alpha, \beta) \leftarrow (r, s, t)$ ■

2.2. El anillo de enteros módulo m, $\mathbb{A}_m$.

Una relación fundamental para la aritmética del anillo de enteros $\mathbb{Z}$, se da en la siguiente definición.

Definición 2.2.1. Dado m un entero mayor que uno, se define la relación *congruencia módulo* m, $\equiv\ (mod\, m)$, como sigue:

$$(\forall a, b \in \mathbb{Z})(a \equiv b\ (mod\, m) \iff m|(b - a)$$

■

La relación así definida, resulta ser una relación de equivalencia, *i.e.*, se cumple el siguiente resultado.

Lema 2.2.1. Dado $m > 1$, la relación de congruencia cumple:

1. $(\forall a \in \mathbb{Z})(a \equiv a\ (mod\, m))$
2. $(\forall a, b \in \mathbb{Z})(a \equiv b\ (mod\, m) \Rightarrow b \equiv a\ (mod\, m))$
3. $(\forall a, b, c \in \mathbb{Z})(a \equiv b\ (mod\, m)\ \&\ b \equiv c\ (mod\, m) \Rightarrow a \equiv c\ (mod\, m))$ ■

Del hecho de que la congruencia módulo m, sea una relación de equivalencia, se sigue que se puede definir su *conjunto cociente*, lo que se hace en el siguiente resultado.

Lema 2.2.2. Dado $m > 1$, para cualquier $a \in \mathbb{Z}$, se define su *clase de equivalencia* como el conjunto.

$$[a]_m = \{x \in \mathbb{Z} : x \equiv a\ (mod\, m)\}$$

los conjuntos así definidos cumplen las propiedades:

1. Si $a = q \cdot m + r$, con $0 \leq r < m$, entonces, $[a]_m = [r]_m = \{u \cdot m + r \,:\, u \in \mathbb{Z}\}$
2. $(\forall a \in \mathbb{Z})(a \in [a]_m)$
3. $(\forall a, b \in \mathbb{Z})([a]_m \cap [b]_m \neq \emptyset \Rightarrow [a]_m = [b]_m)$
4. $\mathbb{Z} = \bigcup_{r=0}^{m-1} [r]_m$

De hecho los conjuntos de la última igualdad, son ajenos dos a dos, y a la familia:

$$\{[0]_m, [1]_m, \ldots, [m-1]_m\}$$

se le llama conjunto cociente y se denota usualmente $\mathbb{Z}/\equiv_m$. ■

Al conjunto cociente, que nosotros denotaremos ahora: $\mathbb{A}_m$, se le puede dotar de una estructura algebraica definiendo sobre él las operaciones de suma y producto de la siguiente manera:

$$(\forall [a], [b] \in \mathbb{A}_m)([a] + [b] = [a + b])$$

$$(\forall [a], [b] \in \mathbb{A}_m)([a] \cdot [b] = [a \cdot b])$$

Cuando del contexto quede claro cual es el valor m que define el módulo, se puede omitir el subíndice para las clases, como se hizo en la defínción de las operaciones anteriores; éstas operaciones están bien definidas, en el sentido de que no dependen de los representantes, pero aún más $\mathbb{A}_m$ con estas operaciones es un anillo conmutativo con unidad, es decir, para cualesquiera $[a], [b], [c] \in \mathbb{A}_m$ se cumplen las siguientes propiedades:

Cerradura de la suma.	Cerradura del producto.
$[a] + [b] \in \mathbb{A}_m$	$[a] \cdot [b] \in \mathbb{A}_m$
Conmutatividad de la suma.	Conmutatividad del producto.
$[a] + [b] = [b] + [a]$	$[a] \cdot [b] = [b] \cdot [a]$
Asociatividad de la suma.	Asociatatividad del producto.
$([a] + [b]) + [c] = [a] + ([b] + [c])$	$([a] \cdot [b]) \cdot [c] = [a] \cdot ([b] \cdot [c])$
Existencia del neutro aditivo.	Existencia del neutro multiplicativo.
$(\exists [0] \in \mathbb{A}_m)([a] + [0] = [a])$	$(\exists [1] \in \mathbb{A}_m)([a] \cdot [1] = [a])$

Existencia de inversos aditivos.
$(\exists [m-a] \in \mathbb{A}_m)([a] + [m-a)] = [0])$
Distributividad de la suma respecto del producto.
$([a] + [b]) \cdot [c] = [a] \cdot [c] + [b] \cdot [c]$

A continuación se presenta un conjunto de ejemplos para diversos módulos.

Ejemplo 2.2.1. Para $m = 11$, se tienen las siguientes tablas de la suma y el producto.

$+$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$
$[0]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$
$[1]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$
$[2]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$
$[3]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$
$[4]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$
$[5]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$
$[6]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$
$[7]$	$[7]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$
$[8]$	$[8]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$
$[9]$	$[9]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$
$[10]$	$[10]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$

$\cdot$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$
$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$	$[0]$
$[1]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$
$[2]$	$[0]$	$[2]$	$[4]$	$[6]$	$[8]$	$[10]$	$[1]$	$[3]$	$[5]$	$[7]$	$[9]$
$[3]$	$[0]$	$[3]$	$[6]$	$[9]$	$[1]$	$[4]$	$[7]$	$[10]$	$[2]$	$[5]$	$[8]$
$[4]$	$[0]$	$[4]$	$[8]$	$[1]$	$[5]$	$[9]$	$[2]$	$[6]$	$[10]$	$[3]$	$[7]$
$[5]$	$[0]$	$[5]$	$[10]$	$[4]$	$[9]$	$[3]$	$[8]$	$[2]$	$[7]$	$[1]$	$[6]$
$[6]$	$[0]$	$[6]$	$[1]$	$[7]$	$[2]$	$[8]$	$[3]$	$[9]$	$[4]$	$[10]$	$[5]$
$[7]$	$[0]$	$[7]$	$[3]$	$[10]$	$[6]$	$[2]$	$[9]$	$[5]$	$[1]$	$[8]$	$[4]$
$[8]$	$[0]$	$[8]$	$[5]$	$[2]$	$[10]$	$[7]$	$[4]$	$[1]$	$[9]$	$[6]$	$[3]$
$[9]$	$[0]$	$[9]$	$[7]$	$[5]$	$[3]$	$[1]$	$[10]$	$[8]$	$[6]$	$[4]$	$[2]$
$[10]$	$[0]$	$[10]$	$[9]$	$[8]$	$[7]$	$[6]$	$[5]$	$[4]$	$[3]$	$[2]$	$[1]$

■

Ejemplo 2.2.2. Para $m = 12$, se tienen las siguientes tablas de la suma y el producto.

$+$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$
$[0]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$
$[1]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$
$[2]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$
$[3]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$
$[4]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$
$[5]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$
$[6]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$
$[7]$	$[7]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$
$[8]$	$[8]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$
$[9]$	$[9]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$
$[10]$	$[10]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$
$[11]$	$[11]$	$[0]$	$[1]$	$[2]$	$[3]$	$[4]$	$[5]$	$[6]$	$[7]$	$[8]$	$[9]$	$[10]$

$\cdot$	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]	[0]
[1]	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]
[2]	[0]	[2]	[4]	[6]	[8]	[10]	[0]	[2]	[4]	[6]	[8]	[10]
[3]	[0]	[3]	[6]	[9]	[0]	[3]	[6]	[9]	[0]	[3]	[6]	[9]
[4]	[0]	[4]	[8]	[0]	[4]	[8]	[0]	[4]	[8]	[0]	[4]	[8]
[5]	[0]	[5]	[10]	[3]	[8]	[1]	[6]	[11]	[4]	[9]	[2]	[7]
[6]	[0]	[6]	[0]	[6]	[0]	[6]	[0]	[6]	[0]	[6]	[0]	[6]
[7]	[0]	[7]	[2]	[9]	[4]	[11]	[6]	[1]	[8]	[3]	[10]	[5]
[8]	[0]	[8]	[4]	[0]	[8]	[4]	[0]	[8]	[4]	[0]	[8]	[4]
[9]	[0]	[9]	[6]	[3]	[0]	[9]	[6]	[3]	[0]	[9]	[6]	[3]
[10]	[0]	[10]	[8]	[6]	[4]	[2]	[0]	[10]	[8]	[6]	[4]	[2]
[11]	[0]	[11]	[10]	[9]	[8]	[7]	[6]	[5]	[4]	[3]	[2]	[1]

■

Observación 2.2.1. De los ejemplos anteriores, se pueden ver los siguientes hechos:

- En la multiplicación módulo 12, se tiene por ejemplo que $[4][6] = [0]$, a pesar de que $[4], [6] \neq [0]$, cuando esto pasa en un anillo se dice que el anillo es un anillo de divisón, y si a es un elemento que cumpla $a \cdot b = 0$, con $b \neq 0$, a se dice divisor de cero.
- En $\mathbb{A}_{12}$, los elementos $[1], [5], [7]$ y $[11]$ son invertibles, mientras que $[0], [2], [3], [4], [6], [8], [9]$ y $[10]$ son divisores de cero.
- En $\mathbb{A}_{11}$, todos los elementos diferentes de $[0]$ son invertibles, y sólo el $[0]$ es divisor de cero.
- En ambos casos puede verse que $[a]$ es invertible si y sólo si a es primo relativo con el módulo, y que todo elemento es o bien invertible o divisor de cero.

■

2.3. El campo de enteros módulo p, $\mathbb{F}_p$.

Como se puede ver de la observación anotada al final de la sección anterior, si p es un número primo, entonces, todos los elementos diferentes de cero son invertibles módulo p, ésto hace de $\mathbb{A}_p$ un campo finito con p elementos, el cuál lo denotaremos como $\mathbb{F}_p$.

La aritmética definida sobre los campos y anillos finitos resulta de gran utilidad para las aplicaciones en criptografía, en este trabajo nos restrigiremos a los anillos de enteros módulo m, $\mathbb{A}_m$ y a los campos $\mathbb{F}_p$ con p primo, si bien se sabe que para cada primo p y cada natural $n > 1$, existe un campo con p^n elementos, a saber el campo $\mathbb{F}_{p^n}$, y que de hecho éstos últimos, salvo isomorfismos, son todos los campos finitos.

A los elementos invertibles de un anillo también se les conoce como unidades, y es usual denotar como A^* al conjunto de unidades del anillo A, como puede verse A^* es un grupo respecto de la operación producto, un anillo A será un campo si y sólo si se tiene $A^* = A - \{0\}$.

En el caso de los anillos $\mathbb{A}_m$, se tiene:

$$\mathbb{A}_m^* = \{[a] : mcd(a, m) = 1\}$$

En efecto del lema (2.1.1), se puede ver que si $mcd(a, m) = 1$, entonces existen α y β, tales que:

$$\alpha \cdot a + \beta \cdot m = 1$$

de donde $[\alpha] \cdot [a] = [1]$ y $[a]^{-1} = [\alpha]$.

Por otro lado si existe $[u]$ tal que $[u]\cdot[a] = [1]$, entonces, $m|(u\cdot a-1)$, *i.e.*, $u\cdot a-1 = q\cdot m$, luego, $u \cdot a - q \cdot m = 1$, y del lema (2.1.2), se sigue que $mcd(a, m) = 1$.

A continuación se presenta un ejemplo donde se muestra como se puede usar el algoritmo (2.1.2), para calcular los inversos multiplicativos dentro de los anillos $\mathbb{A}_m$ y los campos $\mathbb{F}_p$.

Ejemplo 2.3.1. Se aplica el algoritmo (2.1.2) para calcular el máximo común divisor, d, de $a = 2853$ y $b = 347$, así como los coeficientes α y β, tales que $\alpha \cdot a + \beta \cdot b = d$

r	r'	r''	s	t	s'	t'	q	$sa + tb$
2853	347	77	1	0	0	1	8	2853
347	77	39	0	1	1	−8	4	347
77	39	38	1	−8	−4	33	1	77
39	38	1	−4	33	5	−41	1	39
38	1	0	5	−41	−9	74	38	38
1	0	0	−9	74	347	−2853	1	1

Obteniéndose $d = 1$, $\alpha = -9$ y $\beta = 74$, donde

$$1 = mcd(2853, 347) = (-9)2853 + (74)347$$

Ahora como 347 es primo y $2853 \equiv 77\,(mod\,347)$, se tiene que $[77]^{-1} = [-9] = [338]$ en $\mathbb{F}_{347}$, mientras que $[347]^{-1} = [74]$ en $\mathbb{A}_{2853}$. ■

En los algoritmos presentados en los capítulos (4) y (5), se usa de manera fundamental el álgebra, aquí presentada, sobre los anillos y campos finitos, otras técnicas de gran utilidad en otro titpo de aplicaciones criptográficas están relacionadas con el problema del logaritmo discreto, a continuación se define el concepto de generador de un grupo, el cual es la base del planteamiento del problema antes mencionado, después de la definición se enuncia un resulato fundamental para el grupo de unidades de $\mathbb{F}_p$.

Definición 2.3.1. Si $(G, \cdot)$, es un grupo, entonces, un elemento $g_0 \in G$ se dice un generador del grupo, si se cumple:

$$G = \{g_0^t : t \in \mathbb{Z}\}$$

en este caso se dice que G es cíclico. ■

Lema 2.3.1. El grupo de unidades $\mathbb{F}_p^*$, es cíclico, *i.e.*, existe $u_0 \in \mathbb{F}_p^*$ generador del grupo. ■

Cerramos esta sección anotando que el problema del logaritmo discreto se basa en que mientras es fácil calcular $u_t = u_0^t$, para $t \in \mathbb{N}$, en $\mathbb{F}_p^*$, recuperar u_0 de dicho valor es mucho más dificil, al cálculo de tal valor es al que se conoce como el cálculo del logaritmo discreto de u_t.

Capítulo 3

Álgebra lineal.

Se denomina álgebra a la rama de las matemáticas que se orienta a la generalización de las operaciones aritméticas a través de signos, letras y números. En el álgebra, las letras y los signos representan a otra entidad a través de un simbolismo.

Lineal, por su parte, es un adjetivo que refiere a lo vinculado a una línea (una raya o una sucesión). En el ámbito de la matemática, la idea de lineal alude a aquello que cuenta con consecuencias que son proporcionales a una causa

Se conoce como álgebra lineal a la especialización del álgebra que trabaja con matrices, vectores, espacios vectoriales y ecuaciones de tipo lineal. Se trata de un área del conocimiento que se desarrolló especialmente en la década de 1840 con los aportes del alemán Hermann Grassmann (1809-1877) y el irlandés William Rowan Hamilton (1805-1865), entre otros matemáticos.[7]

3.1. Definiciones y propiedades básicas.

3.1.1. Matrices

Sea K el cuerpo de los números reales o números complejos (escalares) y sean m, n números naturales. Una matriz $n \times m$ es una aplicación

$$A : \{1, 2, \cdots, n\} \times \{1, 2, \cdots, m\} \to \mathbb{K}$$

es decir, dados $(i, j) \in \{1, 2, ..., n\} \times \{1, 2, ..., m\}$, $A(i, j) = a_{ij}$ será un número real o complejo. Como el dominio de la matriz A es finito, es más usual escribir una matriz de la siguiente:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{pmatrix}$$

Se dirá entonces que la matriz A tiene n filas y m columnas. Denotaremos por $M_{n\times m}\mathbb{K}$ al conjunto de matrices de tamaño $n \times m$ con coeficientes en $\mathbb{K}$, si está claro del contexto

el orden de una matriz, ésta puede representarse simplemente como (a_{ij}) donde cada elemnto a_{ij} representa al que se encuentra en la fila i y columna j de nuestra matriz. En caso de que $n = m$, es decir, tenemos el mismo número de filas que de columnas, diremos que la matriz A es cuadrada. Por ejemplo, la matriz:

$$A = \begin{pmatrix} 5 & 8 & 4 & 5 \\ 2 & 2 & 1 & 2 \\ 6 & 4 & 0 & 9 \\ 4 & 7 & 5 & 2 \end{pmatrix}$$

3.1.2. Operaciones con Matrices

Suma

Definición 3.1.1. Dadas dos matrices de orden $n \times m$, $A = (a_{ij})$ y $B = (b_{ij})$, se define su suma como:

$$A + B = (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij})$$

Por ejemplo:

$$\begin{pmatrix} 0 & 1 & -2 \\ 4 & 3 & 4 \\ 9 & -4 & -4 \\ 1 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 & -5 \\ 1 & 5 & 8 \\ 2 & -1 & -1 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 2 & -7 \\ 5 & 8 & 12 \\ 11 & -5 & -5 \\ 2 & 2 & 3 \end{pmatrix}$$

Como se ve de la definición, sólo se pueden sumar matrices del mismo orden. *square*

Observación 3.1.1. La suma de matrices tiene las siguientes propiedades:

1. Propiedad asociativa. Dadas A,B,C $\in M_{n\times m}(\mathbb{K})$ se verifica que $(A + B) + C = A + (B + C)$. Para demostrar la propiedad se considera

$$(A + B) + C = ((a_{ij}) + (b_{ij})) + (c_{ij}) = (a_{ij} + b_{ij}) + (c_{ij}) = ((a_{ij} + b_{ij}) + c_{ij})$$

$$= (a_{ij} + (b_{ij} + c_{ij})) = (a_{ij}) + (b_{ij} + c_{ij}) = (a_{ij}) + ((b_{ij}) + (c_{ij}))$$

$$= A + (B + C)$$

2. Propiedad conmutativa. Dadas A,B $\in M_{n\times m}(\mathbb{K})$ se verifica que $A + B = B + A$. Para demostrar la propiedad se considera

$$A + B = (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij}) = (b_{ij} + a_{ij}) = B + A$$

3. Elemento neutro. Se trata de la matriz $0 \in M_{n\times m}(\mathbb{K})$, que es aquella que tiene cero en todas sus componentes. Es claro entonces que dada $A \in M_{n\times m}(\mathbb{K})$ se verifica que

$$A + 0 = 0 + A = A$$

4. Elemento simétrico. Dado $A \in M_{n\times m}(\mathbb{K})$ existe un elemento $-A$ de manera que $A + (-A) = (-A) + A = 0$. Dada la matiz $A = (a_{ij})$ se tiene que $-A = (-a_{ij})$. Entonces es claro que
$$A + (-A) = (a_{ij}) + (-a_{ij}) = 0$$

Por verificarse estas cuatro propiedades, se dice que el par formado por el conjunto de matrices con la operación suma $(M_{n\times m}\mathbb{K}, +)$ es un grupo conmutativo. ■

Producto de matrices

Definición 3.1.2. Dadas las matrices $A \in M_{n\times m}\mathbb{K}$ y $B \in M_{m\times t}\mathbb{K}, m, n, t, \in N$, se define su producto como sigue:

$$A \cdot B = (p_{ij}) \text{ con } p_{ij} = \sum_{k=1}^{m} a_{ik}b_{kj}$$

En consecuencia, para poder multiplicar dos matrices el número de columnas de la primera debe de coincidir con el número de filas de la segunda. Por ejemplo:

$$\begin{pmatrix} 1 & 0 & 2 \\ -1 & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 2 & 0 & 2 \\ 3 & 3 & -3 \end{pmatrix} = \begin{pmatrix} 7 & 7 & -5 \\ -5 & -7 & 7 \end{pmatrix}$$

Algunas propiedades que cumple el producto de matrices son las siguientes.

1. Propiedad asociativa. Dadas $A \in M_{n\times m}\mathbb{K}$,$B \in M_{m\times t}\mathbb{K}$,$C \in M_{t\times l}\mathbb{K}$ se verifica que $(A \cdot B) \cdot C = A \cdot (B \cdot C)$. La demostración es:

$$(A \cdot B) \cdot C = ((a_{ij}) \cdot (b_{ij})) \cdot (c_{ij}) = (\sum_{r=1}^{m} a_{ir}b_{rj}) \cdot (c_{ij})$$

$$= (\sum_{s=1}^{k}(\sum_{r=1}^{m} a_{ir}b_{rs})c_{sj}) = (\sum_{r=1}^{m} a_{ir}(\sum_{s=1}^{k} b_{rs}c_{sj}))$$

$$= (a_{ij}) \cdot (\sum_{s=1}^{k} b_{rs}c_{sj}) = A \cdot (B \cdot C)$$

2. En general no cabe plantearse si se cumple la propiedad conmutativa ya que como vemos en el ejemplo anterior, no se puede hacer el producto en orden inverso porque el número de columnas de la segunda matriz no coincide con el número de filas de la primera matriz. Ahora bien, en caso de poder realizarse el producto, por ejemplo si las matrices son cuadradas la propiedad conmutativa no se verifica como se pone de manifiesto el siguiente ejemplo:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 \\ 3 & 1 \end{pmatrix}$$

mientras que se tiene lo siguiente

$$\begin{pmatrix} 2 & 0 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 2 \\ 2 & 2 \end{pmatrix}$$

3. Existe un elemento neutro, llamado identidad y que es la matriz diagonal que tiene 1 en cada elemento de la diagonal principal y cero fuera de ella. Si llamamos $I_n \in M_{n\times n}\mathbb{K}$ a la matriz identidad y $A \in M_{n\times m}\mathbb{K}$, se verifica que $I_n \cdot A = A \cdot I_m = A$. Si la matriz A es cuadrada, entonces $I_n \cdot A = A \cdot I_n = A$.

4. No toda matriz cuadrada A no nula tiene matriz inversa, es decir, otra matriz A_1 tal que $A \cdot A_1 = A_1 \cdot A = I_n$. En caso de existir tal matriz A_1, diremos que la matriz A es invertible y que A_1 es su matriz inversa y la denotaremos como A^{-1}. Por ejemplo la matriz

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$$

no tiene inversa ya que si existiera tendría que verificarse que

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} a & b \\ a & b \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

y tendríamos que $a = 1$ y $a = 0$ lo cual es imposible. Por ejemplo la matriz siguiente si tiene inversa.

$$\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \end{pmatrix}$$

5. Propiedad distributiva del producto respecto de la suma de matrices. Dadas $A \in M_{n\times m}\mathbb{K}$ y $B, C \in M_{m\times k}\mathbb{K}$, se verifica que $A\cdot(B+C) = A\cdot B + A\cdot C$. Para demostrar esta identidad se considera:

$$A \cdot (B + C) = a_{ij} \cdot (b_{ij} + c_{ij}) = (\sum_{r=1}^{m} a_{ir}(b_{rj} + c_{rj}))$$

$$= \left((\sum_{r=1}^{m} a_{ir}b_{rj}) + (\sum_{r=1}^{m} a_{ir}c_{rj}) \right) = (a_{ij}) \cdot (b_{ij}) + (a_{ij}) + (c_{ij})$$

$$= A \cdot B + A \cdot C$$

Multiplicación de una matriz por un escalar

Definición 3.1.3. Sean $\alpha \in K$ y $A \in M_{n\times m}\mathbb{K}, n, m \in N$. Se define la multiplicación de α por $A = (a_{ij})$ como la matriz $\alpha \cdot A = (\alpha a_{ij})$.

Por ejemplo:

$$2 \cdot \begin{pmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 4 \\ 4 & 6 \\ 6 & 8 \end{pmatrix}$$

Las propiedades que tiene este producto son:

1. Propiedad distributiva del producto respecto de la suma de matrices. Sean $\alpha \in K$ y $A, B \in M_{n\times m}\mathbb{K}$, entonces se verifica que $\alpha \cdot (A + B) = \alpha \cdot A + \alpha \cdot B$. Para probar esta igualdad se considera

$$\alpha \cdot (A + B) = \alpha \cdot ((a_{ij}) + (b_{ij})) = \alpha \cdot (a_{ij} + b_{ij}) = (\alpha(a_{ij} + b_{ij}))$$

$$= (\alpha a_{ij} + \alpha b_{ij}) = (\alpha a_{ij}) + (\alpha b_{ij}) = \alpha(a_{ij}) + \alpha(b_{ij})$$

$$= \alpha \cdot A + \alpha \cdot B$$

2. Propiedad distributiva del producto respecto de la suma de escalares. Sean $\alpha, \beta \in K$ y $A \in M_{n\times m}\mathbb{K}$, entonces se cumple que $(\alpha + \beta) \cdot A = \alpha \cdot A + \beta \cdot A$. Para demostrar la igualdad se toma:

$$(\alpha + \beta) \cdot A = (\alpha + \beta) \cdot (a_{ij}) = ((\alpha + \beta)a_{ij}) = (\alpha a_{ij} + \beta a_{ij})$$

$$= (\alpha a_{ij}) + (\beta a_{ij}) = \alpha \cdot (a_{ij}) + \beta \cdot (a_{ij}) = \alpha \cdot A + \beta \cdot B.$$

3. Propiedad pseudoasociativa.Sean $\alpha, \beta \in K$ y $A \in M_{n\times m}\mathbb{K}$, entonces se cumple que $(\alpha\beta) \cdot A = \alpha \cdot (\beta \cdot A)$. Para demostrar la igualdad sea

$$(\alpha\beta) \cdot A = (\alpha\beta) \cdot (a_{ij}) = ((\alpha\beta)a_{ij}) = (\alpha(\beta a_{ij}))$$

$$= \alpha \cdot (\beta a_{ij}) = \alpha \cdot (\beta \cdot (a_{ij})) = \alpha \cdot (\beta \cdot A).$$

4. Para toda matriz $A \in M_{n\times m}\mathbb{K}$ se verifica que $1 \cdot A = A$. Para demostrarlo consideramos
$$1 \cdot A = 1 \cdot (a_{ij}) = (1 \cdot a_{ij}) = (a_{ij}) = A.$$

Estas propiedades junto con las propiedades de la suma de matrices hace que la terna $(M_{n\times m}\mathbb{K}, +, \cdot)$ tenga estructura de espacio vectorial.

3.1.3. Determinante de matrices

Los determinantes nos permiten saber la compatibilidad de los sistemas de ecuaciones lineales, también permiten la obtención de la solución en el caso de sistemas compatibles determinados.

También tienen otras aplicaciones, por ejemplo: podemos usarlos para determinar si un conjunto de vectores son *linealmente independientes*, y por tanto, forma *una base* del espacio vectorial, vease la sección (3.2); para obtener ecuaciones de planos, superfícies, rectas, etc.

Para el cálculo de determinantes de matrices de cualquier orden, existe una regla recursiva (teorema de Laplace) que reduce el cálculo a sumas y restas de varios determinantes de un orden inferior. Este proceso se puede repetir tantas veces como sea necesario hasta reducir el problema al cálculo de múltiples determinantes de orden tan pequeño como se quiera. Sabiendo que el determinante de un escalar es el propio escalar, es posible calcular el determinante de cualquier matriz aplicando dicho teorema.

Además de esta regla, para calcular determinantes de matrices de cualquier orden podemos usar otra definición de determinante conocida como Fórmula de Leibniz. La fórmula de Leibniz para el determinante de una matriz cuadrada A de orden n es:

$$det(A) = \sum_{\mu \in P_n} sgn(\mu) \prod_{i=1}^{n} a_{i,\mu_i}$$

Donde la suma se calcula sobre todas las permutaciones μ del conjunto $\{1, 2, ..., n\}$, y $sgn(\mu)$ es el signo de la permutación μ, que es igual a ± 1 respectivamente si la permutación es par o impar. La posición del elemento i después de la permutación μ se denota como μ_i. El conjunto de todas las permutaciones es P_n.

La fórmula de Leibniz es útil como definición de determinante; pero, excepto en casos muy pequeños, no es una forma práctica de calcularlo: hay que llevar a cabo $n!$ productos de n factores y sumar $n!$ elementos. No se suele usar para calcular el determinante si la matriz tiene más de tres filas.

Matrices de orden inferior

El caso de matrices de orden inferior (orden 1, 2 ó 3) es tan sencillo que su determinante se calcula con sencillas reglas conocidas. Dichas reglas son también deducibles del teorema de Laplace.

Una matriz de orden uno, es un caso trivial, pero lo trataremos para completar todos los casos. Una matriz de orden uno puede ser tratada como un escalar, pero aquí la consideraremos una matriz cuadrada de orden uno:

$$A = [a_{11}]$$

El valor del determinante es igual al único término de la matriz:

$$detA = det[a_{11}] = |a_{11}| = a_{11}$$

El determinante de una matriz de orden 2:

$$A = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = |A| = a_{11}a_{22} - a_{12}a_{21}$$

El determinante de una matriz de orden 3:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

El determinante se calcula de la siguiente forma:

$$|A| = a_{11}a_{22}a_{33} + a_{12}a_{21}a_{31} + a_{13}a_{21}a_{32} - (a_{31}a_{22}a_{13} + a_{32}a_{23}a_{11} + a_{33}a_{21}a_{12})$$

Ejemplo 3.1.1. Matriz 3×3

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 0 & 2 & -5 \\ -2 & 1 & 4 \end{pmatrix}$$

Calculando determinante:

$$|A| = (3 \cdot 2 \cdot 4) + (2 \cdot (-5) \cdot (-2)) + (1 \cdot 0 \cdot 1) - (-1 \cdot 2 \cdot (-2)) - (2 \cdot 0 \cdot 4) - (3 \cdot (-5) \cdot 1) =$$

$$24 + 20 + 0 - (-4) - 0 - (-15)$$

$$44 + 4 + 15 = 63$$

Determinante igual a 63. ■

3.1.4. Matrices Inversas

Como se anotó al definir el producto de matrices, para algunas existe su matriz inversa, a continuación se presenta formalmente la definición correspondiente.

Definición 3.1.4. Una matriz cuadrada A de orden n se dice que es invertible, no singular, no degenerada o regular si existe otra matriz cuadrada de orden n, llamada matriz inversa de A y representada como A^{-1}, tal que: $A \cdot A^{-1} = A^{-1} \cdot A = I_n$, donde I_n es la matriz identidad de orden n.

Una matriz no invertible se dice que es singular o degenerada. Una matriz es singular si y solo si su *determinante* es nulo, en consecuencia, una matriz es no singular si y solo si su determinante es diferente de cero. La inversión de matrices es el proceso de encontrar la matriz inversa de una matriz dada.

Propiedades de la matriz inversa

1. La inversa de una matriz, es única.
2. La inversa del producto de dos matrices es el producto de las inversas cambiando el orden:
$$(A \cdot B)^{-1} = B^{-1} \cdot A^{-1}$$
3. Si la matriz es invertible, también lo es su *transpuesta* ver subsección (3.1.5), y el inverso de su transpuesta es la transpuesta de su inversa, es decir:
$$(A^t)^{-1} = (A^{-1})^t$$

4. Evidentemente se tiene que:
$$(A^{-1})^{-1} = A$$

5. Una matriz es invertible si y sólo si el determinante de A es distinto de cero. Además la inversa satisface la igualdad:
$$A^{-1} = \frac{1}{|A|} adj(A)^t$$
donde $|A|$ es el determinante de A y $adj(A)$ es la matriz adjunta de A, entendida como a la matriz de cofactores.

Demostración de la unicidad de la inversa.

Supongamos que B y C son inversas de A, *i.e.*,

$$AB = BA = I \; y \; AC = CA = I$$

Ahora multiplicando BA por C, y usando las relaciones anteriores, se tiene

$$(BA)C = IC = C \; y \; (BA)C = B(AC) = B$$

De modo que $B = C$ y se prueba que la inversa es única.

Demostración del criterio de inversibilidad de las matrices cuadradas

Se probara la doble implicación.
Suficiencia ($\rightarrow$)
Suponiendo que A es invertible, *i.e.* que existe B, tal que $AB = BA = I$, al aplicar la funcion determinante se tiene que:

$$det(AB) = det(BA) = det(I)$$

usando que $det(I) = 1$ y que $det(AB) = det(A)det(B)$

$$det(A)det(B) = 1$$

Por lo tanto, $det(A)$ es distinto de cero.

$$det(A) \neq 0$$

Necesidad ($\leftarrow$)

Suponiendo que el determinante de A es distinto de cero, sea a_{ij} el elemento ij de la matriz A y sea A_{ij} la matriz A sin la fila i y la columna j, entonces:

$$det(A) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \; det(A_{ij})$$

Sea $k \neq j$, entonces:

$$\sum_{i=1}^{n}(-1)^{i+j}a_{ik}\ det(A_{ij}) = 0$$

Esta afirmación es válida por propiedades de los determinantes, pues la parte izquierda de la relación es el determinante de la matriz A con la columna j igual a la columna k y los demás términos iguales a los de A. Entonces

$$\delta_{jk}det(A) = \sum_{i=1}^{n}(-1)^{i+j}det(A_{ij})a_{ik}$$

donde $\delta_{jk} = 1$ cuando $j = k$ y $\delta_{jk} = 0$ cuando $j \neq k$, entonces:

$$det(A)I = (adj(A))^t A$$

Es decir que A tiene inversa izquierda

$$\frac{(adj(A))^t}{det(A)}$$

Como $(adj(A))^t = adj(A^t)$, entonces A^t también tiene inversa izquierda que es

$$\frac{(adj(A^t))^t}{det(A^t)} = \frac{adj(A)}{det(A)}$$

Entonces

$$\frac{adj(A)}{det(A)}A^t = I$$

Luego aplicando la transpuesta se tiene

$$A\frac{(adj(A))^t}{det(A)} = I$$

Ejemplo 3.1.2. Dada una matriz 2×2 con determinante no nulo:

$$A^{-1} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{det(A)}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix} = \frac{1}{ad-bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

Está definida siempre y cuando $ad - bc \neq 0$. Así por ejemplo la inversa de la matriz

$$\begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix}^{-1} \rightarrow \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 3 & -1 \\ -5 & 2 \end{pmatrix}$$

ya que se tiene:

$$\begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix}\begin{pmatrix} 3 & -1 \\ -5 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

■

Ejemplo 3.1.3. Dada una matriz 3×3 con determinante no nulo:

$$A^{-1} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}^{-1} = \frac{1}{det(A)} \begin{pmatrix} A & B & C \\ D & E & F \\ G & H & I \end{pmatrix}^{t} = \frac{1}{det(A)} \begin{pmatrix} A & D & G \\ B & E & H \\ C & F & I \end{pmatrix}$$

$$A = (ei - fh) \quad D = -(bi - ch) \quad G = (bf - ce)$$
$$B = -(di - fg) \quad E = (ai - cg) \quad H = -(af - cd)$$
$$C = (dh - eg) \quad F = -(ah - bg) \quad I = (ae - bd)$$

■

3.1.5. Matriz Transpuesta

Definición 3.1.5. Sea A una matriz con m filas y n columnas. La matriz transpuesta, denotada con A^t Está dada por:

$$(a_{ij})^t = (a_{ji}), 1 \leq i \leq n, 1 \leq j \leq m$$

En donde el elemento a_{ij} de la matriz original A se convierte en el elemento a_{ji} de la matriz transpuesta A^t.

Por ejemplo:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{t} = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}^{t} = \begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

Las propiedades de la trasposición de matrices son las siguientes. Sean $\alpha \in K, A, B \in M_{n\times m}\mathbb{K}$ y $C \in M_{m\times l}\mathbb{K}$. Entonces

(a) $(A^t)^t = A$

(b) $(\alpha \cdot A)^t = \alpha \cdot A^t$

(c) $(A + B)^t = A^t + B^t$

(d) $(A \cdot C)^t = C^t \cdot A^t$

(e) Si $A \in M_{n\times n}\mathbb{K}$ es una matriz invertible, entonces, A^t también es invertible, y se tiene: $(A^t)^{-1} = (A^{-1})^t$.

Demostración

La propiedad (a) es inmediata. Para probar (b) se calcula

$$\alpha \cdot A^t = \alpha \cdot (a_{ij})^t = \alpha \cdot (a_{ji}) = (\alpha a_{ji}) = (\alpha \cdot A)^t$$

Para demostrar (c) se calcula

$$A^t + B^t = (a_{ij})^t + (b_{ij})^t = (a_{ji}) + (b_{ji}) = (a_{ji} + b_{ji}) = (a_{ij} + b_{ij})^t = (A + B)^t$$

Para la demostración de (d)

$$C^t \cdot A^t = (c_{ij})^t \cdot (a_{ij})^t = (c_{ji}) \cdot (a_{ji}) = \sum_{k=1}^{n} a_{jk} c_{ki} = (A \cdot C)^t$$

Para demostrar la última propiedad, sea A^{-1} la matriz inversa de A. Entonces $A \cdot A^{-1} = I_n$. Si aplicamos la propiedad anterior se tiene que

$$(A^{-1})^t \cdot A^t = (A \cdot A^{-1})^t = I_n^t = I_n$$

dado que I_n es una matriz diagonal y su transpuesta es ella misma. Ahora bien, tenemos que $(A^{-1})^t \cdot A^t = I_n$, de donde deducimos que A^t es invertible y su matriz inversa es $(A^{-1})^t$.

La matriz traspuesta se utiliza para definir dos tipos especiales de matrices. Una matriz cuadrada A se dice simétrica si $A^t = A$ y se dice antisimétrica si $A^t = -A$. Si una matriz A es simétrica se verifica que

$$(a_{ij}) = A = A^t = (a_{ij})^t = (a_{ji})$$

por lo que se

$$a_{ij} = a_{ji}$$

Ejemplos de matrices simétricas son:

$$\begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 3 \\ 0 & 3 & 0 \end{pmatrix} \quad , \begin{pmatrix} 0 & 1 & 0 & -1 \\ 1 & -1 & 4 & 0 \\ 0 & 4 & 4 & 3 \\ -1 & 0 & 3 & 0 \end{pmatrix}$$

Por el contrario la matriz A es antisimétrica, entonces se tiene que

$$(a_{ij}) = A = -A^t = -(a_{ij})^t = (-a_{ji})$$

por lo que

$$a_{ij} = -a_{ji}$$

y si $i = j$, entonces $2a_{ii} = 0$, por lo que $a_{ii} = 0$. Ejemplos de matrices antisimétricas

$$\begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & -3 \\ 0 & 3 & 0 \end{pmatrix} \quad , \begin{pmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & 4 & 0 \\ 0 & -4 & 0 & 3 \\ 1 & 0 & -3 & 0 \end{pmatrix}$$

3.2. Espacios vectoriales.

Sea K el cuerpo de los números reales o complejos y sea V un conjunto en el que hay definidas una operación interna $+$ de manera que a cada $u, v \in V$ le asocia un elemento $u+v \in V$, y una operación externa $\cdot$ de manera que a cada $\alpha \in K$ y cada $u \in V$ le asocia un elemento $\alpha \cdot u \in V$, cumpliendo las siguientes propiedades para todo $u, v, w \in V$ y para todo $\alpha, \beta \in K$:

1. Propiedad asociativa para la suma: $(u + v) + w = u + (v + w)$.
2. Propiedad conmutativa de la suma: $u + v = v + u$.
3. Existencia de elemento neutro aditivo 0: $0 + u = u$.
4. Para todo $u \in V$ existe elemento inverso o simétrico $-u$ de manera que $u+(-u) = 0$.
5. Propiedad distributiva respecto de la suma en V, es decir, $\alpha \cdot (u + v) = \alpha \cdot u + \alpha \cdot v$.
6. Propiedad distributiva respecto de la suma en K, esto es, $(\alpha + \beta) \cdot u = \alpha \cdot u + \beta \cdot u$.
7. Propiedad pseudoasociativa: $(\alpha\beta) \cdot u = \alpha \cdot (\beta \cdot u)$.
8. $1 \cdot u = u$

Entonces se dice que la terna $(V, +, \cdot)$ tiene estructura de espacio vectorial sobre el cuerpo $\mathbb{K}$, y podemos decir que estas ocho propiedades son los axiomas que definen el concepto de Espacio Vectorial.

De la definición anterior de Espacio Vectorial, se derivan entre otras, las siguientes propiedades:

(a) $0 \cdot u = 0$ para todo $u \in V$.

(b) $\alpha \cdot 0 = 0$ para todo $\alpha \in \mathbb{K}$.

(c) $\alpha \cdot u = 0$ si y solo si $\alpha = 0$ o $u = 0$.

(d) $(-\alpha) \cdot u = -(\alpha \cdot u) = \alpha \cdot (-u)$ para todo $\alpha \in \mathbb{K}$ y todo $u \in V$.

(e) Si $\alpha \cdot u = \alpha \cdot v$ y $\alpha \neq 0$, entonces $u = v$.

(f) Si $\alpha \cdot u = \beta \cdot u$ y $u \neq 0$, entonces $\alpha = \beta$.

(g) $(-\alpha) \cdot (-u) = \alpha \cdot u$ para todo $\alpha \in \mathbb{K}$ y todo $u \in V$.

Demostración

(a) $0 + 0 = 0$ y multiplicando por u tenemos $(0 + 0) \cdot u = 0 \cdot u$ de donde:

$$(0 + 0) \cdot u = 0 \cdot u + 0 \cdot u = 0 \cdot u.$$

Sumando a ambos miembros el inverso de $0 \cdot u$ tenemos que $0 \cdot u = 0$.

(b) Ahora tenemos que $0 + 0 = 0$. Multiplicando por α tenemos $\alpha \cdot (0 + 0) = \alpha \cdot 0$ de donde:

$$\alpha \cdot 0 + \alpha \cdot 0 = \alpha \cdot (0 + 0) = \alpha \cdot 0.$$

Sumando en ambos miembros el inverso de $\alpha \cdot 0$ tenemos que $\alpha \cdot 0 = 0$.

(c) Si $\alpha = 0$ o $u = 0$, por los apartados anteriores tenemos que $\alpha \cdot u = 0$. Supongamos ahora que $\alpha \cdot u = 0$. Si $\alpha = 0$ ya hemos terminado así que supongamos que $\alpha \neq 0$. Entonces multiplicamos por el inverso de α

$$\alpha^{-1} \cdot (\alpha \cdot u) = \alpha^{-1} \cdot 0 = 0$$

y como

$$\alpha^{-1} \cdot (\alpha \cdot u) = (\alpha^{-1}\alpha) \cdot u = 1 \cdot u = u$$

tenemos que $u = 0$.

(d) Por un lado $\alpha + (-\alpha) = 0$ de donde multiplicando por u tenemos que

$$(\alpha + (-\alpha)) \cdot u = 0 \cdot u = 0.$$

Pero por otro lado

$$(\alpha + (-\alpha)) \cdot u = \alpha \cdot u + (-\alpha) \cdot u = 0$$

de donde tenemos que el inverso de $\alpha \cdot u$ es igual a $(-\alpha) \cdot u$, y en consecuencia:

$$-(\alpha \cdot u) = (-\alpha) \cdot u$$

Por otro lado $u + (-u) = 0$. Multiplicando por α ambos miembros y procediendo como en el caso anterior tenemos que

$$-(\alpha \cdot u) = \alpha \cdot (-u)$$

.

(e) Si $\alpha \cdot u = \alpha \cdot v$ y $\alpha \neq 0$, entonces $\alpha \cdot (u - v) = 0$ y por el apartado (c) se tiene que $u - v = 0$, de donde $u = v$.

(f) Si $\alpha \cdot u = \beta \cdot u$ y $u \neq 0$, entonces $(\alpha - \beta) \cdot u = 0$ y por el apartado (c) se tiene que $\alpha - \beta = 0$, de donde $\alpha = \beta$.

(g) Consideramos $(-\alpha) \cdot (-u) = -(\alpha \cdot (-u)) = -(-\alpha \cdot u) = \alpha \cdot u$

■

3.2.1. Subespacios Vectoriales

Definición 3.2.1. Sea V un espacio vectorial sobre $\mathbb{K}$. Un subconjunto $W \subseteq V$, diferente del conjunto vacío, se dice un subespacio vectorial de V si para todo $\alpha, \beta \in \mathbb{K}$ y para todo $u, v \in W$ se verifica que $\alpha \cdot u + \beta \cdot v \in W$. ■

Una primera consecuencia de la definición es que para todo $\alpha \in \mathbb{K}$ y todo $u, v \in W$ se verifican que $u + v \in W$ y $\alpha \cdot u \in W$. Como las operaciones restringidas a W, siguen verificando los ocho axiomas de espacio vectorial, tenemos que W es en si mismo un espacio vectorial sobre $\mathbb{K}$, de donde inferimos que un subespacio vectorial es cualquier subconjunto no vacío de un espacio, que con las mismas operaciones siga siendo espacio vectorial.

Definición 3.2.2. Dados $\{u_1, ..., u_n\} \in V$ se define una combinación lineal de dichos vectores como una expresión de la forma

$$\alpha_1 \cdot u_1 + ... + \alpha_n \cdot u_n$$

donde $\alpha_1, ..., \alpha_n \in \mathbb{K}$ ■

Definición 3.2.3. Dado un subconjunto $S \subset V$ se define el subespacio generado por S como el conjunto de todas las combinaciones lineales finitas de elementos de V. Lo denotamos por $L(S)$, y podemos escribir que:

$$L(S) = \{\alpha_1 \cdot u_1 + ... + \alpha_n \cdot u_n : u_i \in S \ y \ \alpha_i \in \mathbb{K}, i = 1, 2, ..., n\}.$$

■

Observación 3.2.1. Sea $S \subset V$. Entonces el subespacio generado por S, $L(S)$ es un subespacio vectorial de V. ■

Demostración Sean $\alpha, \beta \in \mathbb{K}$ y $u, v \in L(S)$, y veamos que $\alpha \cdot u + \beta \cdot v \in L(S)$. Para ello, sabemos que existen $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_m \in \mathbb{K}$ y $u_1, ..., u_n, v_1, ..., v_m \in S$ tales que $u = \alpha_1 \cdot u_1 + ... + \alpha_n \cdot u_n$ y $v = \beta_1 \cdot v_1 + ... + \beta_m \cdot v_m$. Entonces:

$$\alpha \cdot u + \beta \cdot v = \alpha \cdot (\alpha_1 \cdot u_1 + ... + \alpha_n \cdot u_n) + \beta \cdot (\beta_1 \cdot v_1 + ... + \beta_m \cdot v_m)$$

$$= (\alpha\alpha_1) \cdot u_1 + ... + (\alpha\alpha_n) \cdot u_n + (\beta\beta_1) \cdot v_1 + ... + (\beta\beta_m) \cdot v_m$$

de donde vemos que $\alpha \cdot u + \beta \cdot v$ es una combinación lineal finita de elementos de S y así $\alpha \cdot u + \beta \cdot v$ pertenece a $L(S)$.

Observación 3.2.2. Sean W_1 y W_2 subespacios vectoriales de V. Entonces

(a) $W_1 \cap W_2$ es un subespacio vectorial de V.

(b) $W_1 \cup W_2$ no es un subespacio vectorial de V en general.

(c) $W_1 + W_2$ es un subespacio vectorial de V.
Donde:

$$W_1 + W_2 = \{u + v : u \in W_1, v \in W_2\}$$

3.2.2. Base y dimensión de espacios vectoriales

Dados $u_1, ..., u_n \in V$ se dice que son linealmente independientes (LI) si dada la combinacin lineal

$$\alpha_1 \cdot u_1 + ... + \alpha_n \cdot u_n = 0$$

se verifica que $\alpha_1 = \cdots = \alpha_n = 0$. En caso contrario se dirán linealmente dependientes(LD).

Definición 3.2.4. Un conjunto de vectores $S = \{v_1, v_2, \ldots, v_n\}$ en un espacio vectorial V se denomina base de V si se cumplen las siguientes condiciones.

- S genera a V.
- S es linealmente independiente

Una base posee dos características que se acaban de ver, debe tener suficientes valores para generar a V, pero no tantos de modo que uno de ellos pueda escribirse como una combinación lineal de los demás vectores en S. Si un espacio vectorial consta de una base con un número finito de vectores n, entonces, toda base tiene n elementos y se dice que V es de dimensión finita n. En caso contrario, V es de dimensión infinita.

Ejemplo 3.2.1. Bases

1. La base canónica (o base natural, o base estándar) de $\mathbb{R}^n$:

$$e_1 = (1, 0, \ldots, 0)$$

$$e_2 = (0, 1, \ldots, 0)$$

$$\vdots$$

$$e_n = (0, 0, \ldots, 1)$$

2. Una segunda base de $\mathbb{R}^n$:

$$e_1 = (1, 1, \ldots, 1, 1)$$

$$e_2 = (0, 1, \ldots, 1, 1)$$

$$\vdots$$

$$e_n = (0, 0, \ldots, 0, 1)$$

- Ambas son linealmente independientes porque forman un determinante no nulo.
- Ambas son sistemas generadores de $\mathbb{R}^n$, en particular para la canónica se tiene que todo vector $(a_1, a_2, \ldots, a_n) \in \mathbb{R}^n$ se puede expresar como combinación lineal de ellos:

$$(a_1, a_2, \ldots, a_n) = a_1(1, 0, \ldots, 0) + a_2(0, 1, \ldots, 0) + \ldots + a_n(0, 0, \ldots, 1)$$

■

Como se mencionó antes, y como se ve en el ejemplo anterior, un espacio puede tener diferentes bases, pero todas éstas tiene el mismo número de elementos, a dicho número se le llamó la dimensión del espacio, otra forma de definir dimensión es la siguiente.

Definición 3.2.5. La dimensión es el máximo número de vectores independientes que podemos tener en el espacio o subespacio. En otras palabras, es el máximo rango que puede tener un conjunto de vectores de dicho espacio.

Es también el rango de cualquier sistema generador de dicho espacio.

Observación 3.2.3. Propiedades de dimensión

1. Significado físico de la dimensión: el espacio tiene dimensión 3, los planos dimensión 2, las rectas dimensión 1, el punto dimensión 0. El subespacio 0 es el único de dimensión 0.

2. La dimensión de un subespacio en $\mathbb{R}^n$, coincide con el número de parámetros libres en su forma paramétrica. (1 parámetro=recta, 2 parámetros= plano...)

3. Si S y T son subespacios y S está contenido en T, entonces $dimS \leq dimT$. Además, si se da la igualdad, $dimS = dimT$, entonces ambos espacios han de coincidir.

4. El rango de una familia de vectores, es igual a la dimensión del subespacio que generan. Es decir: si $v_1, v_2, \dots v_n$ generan un cierto subespacio S, y si el rango de dicho conjunto es r, entonces dim $S = r$. (Si un cierto conjunto de vectores tienen rango 2, entonces generan un plano; etc.)

Ejemplo 3.2.2. Dimension

1. $\mathbb{R}^n$ tiene dimensión n, pues tiene una base de n elementos.

2. $M_{2\times 2}$ = matrices 2×2 con términos reales tiene dimensión 4. Una base de $M_{2\times 2}$ es:
$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} , \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} , \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} , \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$$

■

Sea S un espacio o subespacio de dimensión m. Entonces,

- Si tenemos m vectores linealmente independientes en S, también serán sistema generador de S.

- Si tenemos m vectores que generan S, también serán linealmente independientes.

Por tanto, si tenemos un conjunto formado por tantos vectores como indica la dimensión, dichos vectores serán a la vez linealmente independientes y sistema generador, o bien ninguna de las dos cosas.

Así para probar que son base, bastaría probar solamente una de las dos cosas:

Que son linealmente independientes, o que son sistema generador. Esto solamente se puede aplicar cuando conocemos la dimensión del espacio y cuando tenemos tantos vectores como indica la dimensión.

Observación 3.2.4. En un espacio o subespacio de dimensión m,

- un conjunto de más de m vectores nunca puede ser linealmente independiente.
- un conjunto de menos de m vectores nunca puede ser sistema generador. ■

3.2.3. Espacios vectoriales sobre campos finitos.

Como se anotó en la presentación del concepto de espacio vectorial, un espacio vectorial consta de un par de conjuntos: el conjunto de vectores V y un conjunto de escalares K, donde este último es un campo, ahora, si usualmente en los cursos de álgebra lineal, K coincide con el campo de los números reales, y se desarrolla lo que se definiría como álgebra sobre espacios vectorales reales, o en algunos casos se exiende el campor K, al campo de los numeros complejos, haciendo en este caso álgebra sobre espacios vectoriales complejos, en las aplicaciones de criptografía, es importante desarrollar el álgebra lineal sobre campos finitos, en particular sobre los campos $\mathbb{F}_p$, o aún más recuperar algunas de las propiedades de esta álgebra, cuando está definida sólo sobre los anillos $\mathbb{A}_m$, como puede verse en el desarrollo del algoritmo de Hill , afortunadamente esto puede realizarse tal como se muestra en la sección (4.2) cuidando sólo las características particulares del álgebra de los anillos $\mathbb{A}_m$ y los campos $\mathbb{F}_p$ respectivamente, en particular aquí destacamos el siguiente resultado.

Lema 3.2.1. Si A es una matriz cuadrada $n \times n$, con elementos en $\mathbb{A}_m$, entonces, los siguientes enunciados son equivalentes:

- Los vectores fila son linealmente independientes.
- Los vectores columna son linealmente independientes.
- El rango de la matriz A es n.
- La matriz A es invertible.
- El determinante de la matriz $|A|$ es primo relativo con n. ■

Si una matriz A, con elementos en $\mathbb{A}_m$ es invertible, entonces se sigue cumpliendo que su inversa se puede calcular según la siguiente igualdad:

$$A^{-1} = \frac{1}{|A|} adj(A)^t$$

Finalmente vale la pena observar que los cálculos que realizamos en espacios vectoriales reales, usando operaciones elementales de filas o columnas, también pueden realizarse en el caso de estar trabajando sobre los anillos $\mathbb{A}_m$, o los campos $\mathbb{F}_P$, a aún sobre los campos $\mathbb{F}_{p^n}$, en particular con estas operaciones, también se puede determinar si una matriz es invertible, y en caso de serlo calcular su matriz inversa.

Capítulo 4

Criptografía de llave privada.

La aplicación principal de los algoritmos de cifrado es garantizar la confidencialidad de los documentos aunque estos resultasen accesibles a personas no autorizadas. La situación práctica en la que más se utiliza es la transferencia de información por canales de comunicación no seguros, como es Internet.

Las técnicas de cifrado, además de garantizar la confidencialidad, garantizan colateralmente la integridad, ya que si el documento no es accesible para usuarios no autorizados tampoco es modificable. Los algoritmos de cifrado juegan un papel decisivo en la transferencia de archivos, por ejemplo por correo electrónico, y en la transferencia de información mediante navegadores, por ejemplo durante el acceso a la página web de un banco.También se utilizan las técnicas de cifrado para proteger documentos importantes dentro del disco duro o en cualquier medio de almacenamiento digital, por si se produce un acceso ilegal.[3]

La criptografía clásica abarca desde tiempos inmemoriales hasta la mitad del siglo XX. El punto de inflexión en esta clasificación la marcan tres hechos relevantes:

- En el año 1948 se publica el estudio de Claude Shannon sobre la Teoría de la Información.
- En 1974 aparece el estándar de cifra DES.
- Y en el año 1976 se publica el estudio realizado por Whitfield Diffie y Martin Hellman sobre la aplicación de funciones matemáticas de un solo sentido a un modelo de cifrado, denominado de clave pública.

Los criptosistemas clásicos son todos simétricos y los podemos clasificar según el proceso de cifrado en:

- Criptosistemas de transposición: el texto cifrado es el resultado de una reordenación de los símbolos del texto original
- Criptosistemas de sustitución: cada sḿbolo del texto es sustituido por otro símbolo
    - Monoalfabética: cada símbolo del texto se sustituye por otro símbolo que es siempre el mismo, no depende de la posición en el texto

- Polialfabética: cada símbolo del texto se sustituye por otro símbolo dependiendo de la posición en el texto.[6]

Algunos ejemplos de algoritmos simétricos son DES, 3DES, RC5, AES, Hill e IDEA

4.1. Aplicación de los algoritmos simétricos.

El cifrado simétrico, también conocido como cifrado de clave privada o cifrado de clave secreta consiste en utilizar un par de llaves, **k** y **k'**, para el cifrado y el descifrado respectivamente, es decir, si se cifra un mensaje **m** con una llave secreta **k** entonces el mensaje cifrado resultante **m'** únicamente se podrá descifrar con la llave **k'**.

Este tipo de llave conocida como secreta se debe de compartir entre las personas que se desea que vean los mensajes. Con este tipo de criptografía podemos garantizar la confidencialidad porque únicamente quien posea la llave secreta será capaz de ver el mensaje.

La criptografía de clave simétrica tiene varios beneficios. Este tipo de cifrado es muy fácil de usar. Además, es muy útil para el cifrado de archivos de datos personales, ya que sólo se requiere de una clave. La criptografía de clave simétrica es rápida y utiliza menos recursos informáticos que otras formas de cifrado. Esta forma de cifrado también se puede utilizar para ayudar a prevenir riesgos en la seguridad. Si se utilizan diferentes claves compartidas con diferentes personas, cuando una de las claves está en peligro, sólo una persona se ve afectada en lugar de todos.

El problema con la criptografía simétrica es que si se quisiera compartir mensajes secretos con n personas, para cada persona se tendría que generar una nueva llave secreta y la administración personal de todas las n llaves sería un problema. Otro problema asociado con este tipo de criptografía es cómo se comparte con otra persona de una forma confidencial e integra la llave secreta.

Ejemplos en la historia de criptosistemas simétricos son:

La escítala
La escítala era usada en el siglo V a.d.C. por el pueblo griego de los lacedemonios. Consistía en un bastón en el que se enrollaba una cinta de cuero y luego se escribía en ella el mensaje de forma longitudinal. Al desenrollar la cinta, las letras aparecían desordenadas. Para descifrar el criptograma y recuperar el mensaje en claro habría que enrollar dicha cinta en un bastón con el mismo diámetro que el usado en el extremo emisor y leer el mensaje de forma longitudinal. La clave del sistema se encuentra en el diámetro del bastón. Se trata de una cifra por transposición pues los caracteres del criptograma son los mismos que en el texto en claro pero están distribuidos de otra forma dentro del criptograma.[6]

El cifrador del César
En el siglo I a.d.C., Julio César usaba este cifrador. El algoritmo consiste en el desplazamiento de tres espacios hacia la derecha de los caracteres del texto en claro. Es un

cifrador por sustitución monoalfabético en el que las operaciones se realizan módulo n, siendo n el número de elementos del alfabeto.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
D	E	F	G	H	I	J	K	L	M	N	O	P	Q

O	P	Q	R	S	T	U	V	W	X	Y	Z
R	S	T	U	V	W	X	Y	Z	A	B	C

Alfabeto de cifrado del César para castellano mod 26.
Cifrado: Ci = Mi + 3 mod 26
Descifrado: Mi = Ci - 3 mod 26

Ejemplo 4.1.1.

$$M = ELPATIODEMICASAESPARTICULAR$$
$$C = HNSDWLRGHOLFDVDHVSDUWLFXNDU$$

Se observa como cada letra se recorre tres posiciones a la derecha conforme el alfabeto que estamos usando. ■

Cada letra se cifrará siempre igual. Es una gran debilidad y hace que este sistema sea muy vulnerable y fácil de atacar, simplemente usando estadísticas sobre la repetición de las letras de nuestro alfabeto dentro del lenguaje usado.

4.2. Algoritmo Hill.

En 1929 el matemático Lester Hill propone un sistema de cifra usando una matriz como clave, cifrando Ngramas de forma que:

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & ... & k_{1N} \\ k_{21} & k_{22} & k_{23} & ... & k_{2N} \\ k_{31} & k_{32} & k_{33} & ... & k_{3N} \\ ... & ... & ... & & ... \\ k_{N1} & k_{N2} & k_{N3} & ... & k_{NN} \end{pmatrix} \times \begin{pmatrix} M_1 \\ M_2 \\ M_3 \\ \vdots \\ M_N \end{pmatrix} \quad (mod\ n)$$

La matriz clave K debe tener inversa K^{-1} en el cuerpo de cifra n. Luego, como:

$$K^{-1} = (ADJ(K))^t/|K|(mod\ p)$$

en donde ADJ(K) es la matriz adjunta y $|K|$ el determinante, este último valor $|K|$ debe ser invertible en el anillo $\mathbb{A}_n$, y por lo tanto debe cumplirse: $mcd(|K|, n) = 1$.

Si el texto en claro no es múltiplo del tamaño de bloque, N, se rellena con caracteres predeterminados, por ejemplo la letra X o la Z.

Para llevar a cabo una comunicación mediante este sistema de cifrado el emisor y el receptor se deben poner de acuerdo en el tamaño de los bloques del texto N, la matriz que utilizarán como llave y el alfabeto de símbolos que utilizarán. Sea m el mensaje a cifrar, entonces el emisor divide m en bloques de longitud N: $m = m_1||m_2||...||m_k$.

Los elementos de cada bloque m_i en los que se ha dividido el mensaje se convertirán a sus número equivalente módulo 26. El alfabeto utilizado para el módulo 26 es el siguiente:

A	B	C	D	E	F	G	H	I	J	K	L	M	N
	O	P	Q	R	S	T	U	V	W	X	Y	Z	

Una vez que se han convertido los caracteres a sus números equivalentes se crean los vectores columna M_i de longitud N, formados por los elementos de cada bloque m_i.

Para cifrar sólo se necesita una matriz que se utilizara como llave. No todas las matrices pueden ser utilizadas como llave, para ello se deben emplear matrices invertibles. Una matriz es invertible si:

En $\mathbb{R}$	En Z_n
k es invertible $\Leftrightarrow$ det(K) $\neq 0$ $k^{-1} = \frac{1}{det(k)} * (Adjk)^T$	k es invertible $\Leftrightarrow$ $det(k) \neq 0$ módulo n k es invertible $\Leftrightarrow$ det(k) y n coprimos $k^{-1} = (detk)^{-1} * (Adjk)^T$

En este criptosistema, K debe estar formada por coeficientes en $\mathbb{A}_n$, siendo n el número de elementos del alfabeto. En nuestro caso n es 26. El proceso de cifrado es el siguiente:

$$K \cdot M_i = C_i$$

Es decir, si la matriz K (llave) es una matriz cuadrada de orden 3, entonces M_i es una matriz de 3 filas y 1 columna que contiene tres elementos del mensaje original y C_i es una matriz de 3 filas y 1 columna que contiene los tres elementos,correspondientes, ya cifrados.

El proceso de descifrado es el siguiente:

$$M_i = K^{-1} \cdot C_i$$

donde K^{-1} es la matriz inversa de K que es la llave.

Se descifra multiplicando la matriz inversa de la llave por C_i obteniendo M_i, para descifrar el bloque C_i que contiene tres elementos del mensaje cifrado obteniendo así M_i que contiene los tres elementos, correspondientes, del mensaje original.

Ejemplo 4.2.1.

Ejemplo de Algoritmo Hill

Se cifrara el mensaje "codigo" utilizando un alfabeto de 26 caracteres donde cada carácter corresponde a un número. Se utilizara la siguiente matriz cuadrada de orden 3 y con coeficientes en $\mathbb{Z}_{26}$ como clave K y vamos a comprobar que es invertible.

$$K = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix}$$

Se calcula la determinante de la matriz K para ver si es invertible.

$$|K| = \begin{vmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{vmatrix}$$

$$|K| = 503$$

$$5(23 \cdot 13 - 3 \cdot 11) - 17(9 \cdot 13 - 3 \cdot 2) + 20(9 \cdot 11 - 23 \cdot 2) =$$

$$1215 - 1734 + 1060 = 503$$

$$503 = 9 \; mod \; 26 \Rightarrow m.c.d(9, 26) = 1$$

Como el determinante de K y el módulo de $\mathbb{Z}_n$ son coprimos entonces la matriz es invertible.

Para encriptar el mensaje "CODIGO" debemos encriptar los seis caracteres de "CODIGO" en bloques de 3 caracteres cada uno, los bloques quedan como siguen:

$$cod \rightarrow (2, 14, 3) = M_1$$

$$igo \rightarrow (8, 6, 14) = M_2$$

El proceso de cifrado se hará de la siguiente forma:

$$K \cdot M_1 = C_1$$
$$K \cdot M_2 = C_2$$

$$K \cdot M_1 = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix} \begin{pmatrix} 2 \\ 14 \\ 3 \end{pmatrix} = \begin{pmatrix} 308 \\ 349 \\ 197 \end{pmatrix} = \begin{pmatrix} 22 \\ 11 \\ 15 \end{pmatrix} \; (\; mod \; 26) \; \rightarrow \; (w, l, p)$$

$$K \cdot M_2 = \begin{pmatrix} 5 & 17 & 20 \\ 9 & 23 & 3 \\ 2 & 11 & 13 \end{pmatrix} \begin{pmatrix} 8 \\ 6 \\ 14 \end{pmatrix} = \begin{pmatrix} 422 \\ 252 \\ 264 \end{pmatrix} = \begin{pmatrix} 6 \\ 18 \\ 4 \end{pmatrix} \ (\ mod\ 26) \ \rightarrow \ (g, s, e)$$

Luego "CODIGO" encriptado equivale a "WLPGSE".

Observar que las dos "O" se codificaran de forma diferente.

Para hacer el proceso de descifrado tenemos que calcular la matriz inversa de la clave K. La matriz inversa se calcula mediante la fórmula:

$$K^{-1} = (detK)^{-1} \cdot (AdjK)^T$$

$$det\ K \equiv 503 \equiv 9(\ mod\ 26)$$

Resolviendo $9x = 1\ (mod\ 26)$ obtenemos que $x = 3 \Rightarrow (detK)^{-1} = 3$

La matriz Adj(K) se calcula sustituyendo cada elemento de K por sus respectivos adjuntos. El adjunto de un elemento k_{ij} es el resultante de multiplicar $(-1)^{i+j}$ por el determinante de la matriz que se obtiene al quitar de la matriz K la fila y columna que contiene el elemento k_{ij}.

La matriz transpuesta de K se obtiene al intercambiar filas por columnas.

$$Adj(K) = \begin{pmatrix} 266 & -111 & 53 \\ -1 & 25 & -21 \\ -409 & 165 & -38 \end{pmatrix} \ \rightarrow \ (Adj\ K)^T \begin{pmatrix} 266 & -1 & -409 \\ -111 & 25 & 165 \\ 53 & -21 & -38 \end{pmatrix}$$

$$K^{-1} = (detK)^{-1} \cdot (AdjK)^T = 3 \cdot \begin{pmatrix} 266 & -1 & -409 \\ -111 & 25 & 165 \\ 53 & -21 & -38 \end{pmatrix} = \begin{pmatrix} 798 & -3 & -1227 \\ -333 & 75 & 495 \\ 159 & -63 & -114 \end{pmatrix}$$

$$K^{-1} = \begin{pmatrix} 18 & 23 & 21 \\ 5 & 23 & 1 \\ 3 & 15 & 16 \end{pmatrix} \ (\ mod\ 26)$$

Para recuperar el mensaje el proceso de descifrado sería:

$$M_1 = K^{-1} \cdot C_1$$
$$M_2 = K^{-1} \cdot C_2$$

$$K^{-1} \cdot C_1 = \begin{pmatrix} 18 & 23 & 21 \\ 5 & 23 & 1 \\ 3 & 15 & 16 \end{pmatrix} \cdot \begin{pmatrix} 22 \\ 11 \\ 15 \end{pmatrix} = \begin{pmatrix} 964 \\ 378 \\ 471 \end{pmatrix} \rightarrow \begin{pmatrix} 2 \\ 14 \\ 3 \end{pmatrix} \ (\ mod\ 26)$$

$$K^{-1} \cdot C_2 = \begin{pmatrix} 18 & 23 & 21 \\ 5 & 23 & 1 \\ 3 & 15 & 16 \end{pmatrix} \cdot \begin{pmatrix} 6 \\ 18 \\ 4 \end{pmatrix} = \begin{pmatrix} 606 \\ 448 \\ 352 \end{pmatrix} \rightarrow \begin{pmatrix} 8 \\ 6 \\ 14 \end{pmatrix} \quad (\ mod\ 26)$$

$$cod \rightarrow (2, 14, 3) = M_1$$
$$igo \rightarrow (8, 6, 14) = M_2$$

■

Por lo tanto se ha concluido con el proceso de recuperar el mensaje, se puede observar que en efecto el mensaje recuperado es igual que el original.

Capítulo 5

Criptografía de llave pública.

En 1975, dos ingenieros de la Universidad de Stanford, Whitfield Diffie y Martin Hellman, sugieren para el diseño de criptosistemas seguros, usar algoritmos que computacionalmente sean prácticamente irresolubles si no se tiene la información adecuada.

La idéa consistía básicamente en encontrar un sistema de cifrado computacionalmente fácil, de tal manera que el descifrado sea, por el contrario, muy complicada, a menos que se conozca la clave. Para ello se buscan definir funciones invertibles, de tal forma que tanto la función como su inversa sean fáciles de calcular, pero que conociendo la función el cálculo de la inversa sea lo suficientemente complicado como para que resulte en la práctica imposible su cálculo.

La criptografía asimétrica resuelve las dos desventajas principales de la simétrica:

- No se necesita canales seguros para mandar la clave. La distribución de claves es más fácil y segura ya que la clave que se distribuye es la pública manteniéndose la privada para el uso exclusivo del propietario.
- No hay desbordamiento en el tratamiento de claves y canales.

Pero este tipo de algoritmos también tiene desventajas:

- Son poco eficientes. Las claves deben de ser largas y se tarda bastante tiempo en aplicarlas.
- Utilizar las claves privadas repetidamente puede hacer que reciban ataques criptográficos que se basan en analizar paquetes cifrados.
- Hay que proteger la clave privada. Las claves privadas se guardan todas juntas en un archivo llamado keyring, que a su vez está protegido con cifrado simétrico, lo que quiere decir que para usar la clave privada, hay que introducir una clave que descifra el archivo keyring y permita leerla. Esto hace que se necesite una copia de seguridad del llavero keyring.
- Hay que transportar la clave privada. Por eso hay que transportar el llavero con el riesgo que esto supone. Lo mejor para proteger y transportar la clave privada es la tarjeta inteligente. Una tarjeta inteligente (smart card), es cualquier tarjeta de

plástico del tamaño de un bolsillo con circuitos integrados que permiten la ejecución de cierta lógica programada.[4]

De los algoritmos asimétricos se encuentran: Diffie-Hellman, RSA, DSA, ElGamal, Merkle-Hellman ,Knapsack.

5.1. Aplicación de los algoritmos asimétricos.

En estos esquemas se utiliza una clave de cifrado: una llave pública k que determina la función trampa T_k y una llave de descifrado (secreta o privada) que permite el calculo de la inversa T_k^{-1}. Cualquier usuario puede cifrar usando la llave pública, pero soló aquellos que conozcan la llave secreta pueden descifrar correctamente.

Si B, quiere enviar a A, un mensaje cifrado mediante un criptosistema de llave pública debe seguir lo siguiente:

1. B localiza la llave pública de A.
2. B cifra el mensaje haciendo uso de la llave pública de A y le manda el mensaje.

Los pasos a seguir para que A descifre la información recibida son los siguientes:

1. A recibe el mensaje cifrado.
2. A descifra el mensaje aplicando su llave privada, que sólo él conoce.

La fortaleza y seguridad del sistema se basa en que soló A conoce su llave privada, por lo que es la única persona que puede descifrar los mensajes cifrados con su llave pública.

5.1.1. Diffie Helman

El protocolo de Diffie-Hellman fue publicado en 1976. Actualmente se conoce que es vulnerable a ataques de hombre en medio (MitM): un atacante podría situarse entre ambas máquinas y acordar una clave simétrica con cada una de las partes, haciéndose pasar por el host A de cara al host B y viceversa. Una vez establecidas las dos llaves simétricas, el atacante haría de puente entre los dos, descifrando toda la comunicación y volviéndola a cifrar para enviársela al otro. Para corregir la vulnerabilidad del protocolo, éste debe ser utilizado conjuntamente con algún sistema que autentique los mensajes.

Algoritmo 5.1.1. Diffie Hellman

El algoritmo funciona de la siguiente manera:

1. Se usan un par de números, públicos, p un número primo *grande* y g, uno cuyo orden en $\mathbb{F}_p$ sea también grande.

2. Se eligen un par de números a y b, privados, los cuales pueden de ser números no primos; sin embargo, deben también ser números grandes.

3. Se genera una llave pública, a partir del número generador g, del número primo de referencia p y de la llave privada, a y b, según la fórmula siguiente:

$$A \equiv g^a \ (mod\ p) \quad B \equiv g^b \ (mod\ p)$$

4. Finalmente intercambian los valores encontrados, calculando cada uno de los participantes el valor $g^{ab}\ (mod\ p)$, la cual será su llave privada compartida. ■

Ejemplo 5.1.1.

Ejemplo de algoritmo Diffie Hellman

1. Se eligen $p = 941$ y $g = 627$, que se hacen públicos.

2. Cada uno de los dos participantes, elige, respectivamente, $a = 347$ y $b = 781$, y realizan respectivamente los cálculos.

$$A = 390 \equiv 627^{347} \ (mod\ 941) \quad B = 691 \equiv 627^{781} \ (mod\ 941)$$

A y B son las llaves públicas.

3. Se intercambian los valores de A y B, y cada uno cálcula

$$470 \equiv 627^{347 \cdot 381} \ (mod\ 941)$$

Esta es la llave secreta.

■

5.1.2. ElGamal

El procedimiento de cifrado/descifrado ElGamal se refiere a un esquema de cifrado basado en el problema matemático del logaritmo discreto. Es un algoritmo de criptografía asimétrico basado en la idea de Diffie-Hellman y que funciona de una forma parecida a este algoritmo discreto.

El algoritmo de ElGamal puede ser utilizado tanto para generar firmas digitales como para cifrar o descifrar.

Fue descrito por Taher Elgamal en 1984 y se usa en software GNU Privacy Guard, versiones recientes de PGP, y otros sistemas criptográficos. Este algoritmo no está bajo ninguna patente lo que lo hace de uso libre.

La seguridad del algoritmo se basa en la suposición que la función utilizada es de un sólo sentido debido a la dificultad de calcular un logaritmo discreto.

El procedimiento de cifrado (y descifrado) está basado en cálculos sobre un grupo cíclico cualquiera G, lo que lleva a que la seguridad del mismo dependa de la dificultad de calcular logaritmos discretos en G.

Algoritmo 5.1.2. ElGamal
Algoritmo:

1. Si uno de los participantes, elige su llave privada a y pública su llave $A \equiv g^a \ (mod\ p)$
2. Entonces el otro que desea enviarle un mensaje codificado con el entero m, elige una llave temporal k y encripta, y envía, su mensaje en un par (c_1, c_2), donde:
$$c_1 \equiv g^k \ (mod\ p) \quad c_2 \equiv mA^k \ (mod\ p)$$
3. Luego el participante que conoce la llave secreta a, puede calcular $c \equiv c_1^a \ (mod\ p)$, y también $c^{-1} \ (mod\ p)$
4. Finalmente con este valor obtiene
$$\begin{array}{lll} c^{-1} \cdot c_2 & \equiv (c_1^a)^{-1} \cdot c_2 \ (mod\ p) & pues\ c \equiv c_1^a \ (mod\ p) \\ & \equiv (g^{ak})^{-1} \cdot (mA^k) \ (mod\ p) & pues\ c_1 \equiv g^k\ , c_2 \equiv mA^k \ (mod\ p) \\ & \equiv (g^{ak})^{-1} \cdot (m(g^a)^k) \ (mod\ p) & pues\ A \equiv g^a \ (mod\ p) \\ & \equiv m \ (mod\ p) & \end{array}$$

■

Ejemplo 5.1.2.

Ejemplo de algoritmo ElGamal
A elige los valores respectivos.

$$p = 467 \quad g = 2 \quad a = 153$$

A calcula

$$A \equiv g^a \ (mod\ p) \equiv 2^{153} \ (mod\ 467) = 224$$

por lo tanto la llave pública sera:

$$A = 224$$

B escoge $m = 331$ y $k = 197$ y calcula:

$$c_1 \equiv g^k \ (mod\ p) \equiv 2^{197} \ (mod\ 467) = 87$$

$$c_2 \equiv A^k \cdot m \ (mod\ p) \equiv 224^{197} \cdot 331 \ (mod\ 467) = 57$$

Para descifrar A usa la llave privada $a = 153$

$$x \equiv C_1^a \equiv 87^{153} (mod\ 467) = 367\ y\ se\ tiene\ x^{-1} \equiv 4 (mod\ 467)$$

■

5.2. Algoritmo RSA.

Fue desarrollado en 1977 por Ronald Rivest, Adi Shamir y Leonard Adleman, de ahí el nombre de RSA, que corresponde a las iniciales de los apellidos de sus autores. Se basa en criptografía de clave pública y se fundamenta en la dificultad de factorizar en un tiempo razonable cantidades que son producto de dos números primos grandes.

Generación de llaves
Se escogen dos números primos p y q y se calcula $N = p \cdot q$.
Sea ϕ la función phi de Euler, se calcula:

$$\phi(N) = \phi(p \cdot q) = (p-1) \cdot (q-1) = n$$

Se escoge un e tal que $m.c.d(e, n) = 1$, es decir $e \in \mathbb{A}_n^*$
Se calcula $d = e^{-1} \ (mod\ n)$
Por lo tanto las llaves quedan de la siguiente manera:

Clave pública	e	N

Clave privada	d

Cifrado RSA

Se tiene que tener en cuenta que el mensaje m se divide en paquetes de letras de tal forma que su representación numérica sea menor que N, y que debe ser invertible en $\mathbb{A}_n$. El cifrado se realiza de la siguiente manera:

$$c = m^e \ (mod\ N)$$

Descifrado RSA

El descifrado se realiza de la siguiente manera:

$$m = c^d \ (mod\ N)$$

ya que se tiene:

- **Teorema de Euler:** $a^{\phi(N)} \equiv 1 \ (mod\ N) \ \forall \ a \in \mathbb{A}_N^*$
- $m^{\phi(N) \cdot K} \cdot m \equiv (m^{\phi(N)})^k \cdot m \equiv m \ mod\ (N)$
- $e \cdot d \equiv 1 \ (mod\ \phi(N))$
- $e \cdot d = \phi(N) \cdot k + 1$

Y en consecuencia:

$$c^d \equiv (m^e)^d \equiv m^{ed} \ (mod\ N) \equiv m^{\phi(N)k+1} \equiv (m^{\phi(N)})^k \cdot m \equiv m \ (mod\ N)$$

Ejemplo 5.2.1.

Ejemplo de RSA

Cifrar. Se escogen p y q.

$$p = 29 \; q = 31.$$

$$N = p \cdot q = 29 \cdot 31 = 899.$$

$$\phi(N) = \phi(p \cdot q) = n = (p-1)(q-1) = (28-1) \cdot (30-1) = 899$$

Se busca e tal que $m.c.d(e, n) = 1$

$$e = 37 \rightarrow m.c.d(3, 899) = 1$$

Se calcula d tal que $e \cdot d \equiv 1 \; (mod \; 840)$

$$d = 613$$

Quedando como clave pública $(899, 37)$ y la clave privada como (613)
Sea $\sum$ un alfabeto con $K = 26$ letras, donde se ha identificado $A = 1,\ldots,Z = 26$

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	2	3	4	5	6	7	8	9	10	11	12	13	14

O	P	Q	R	S	T	U	V	W	X	Y	Z
15	16	17	18	19	20	21	22	23	24	25	26

Sea m el mensaje, m = "congreso". Utilizando el alfabeto $\sum$ se tiene la siguiente codificación.

C	O	N	G	R	E	S	O
3	15	14	7	18	5	19	15

Los bloques a cifrar son:

$$(3, 15) \;\; (14, 7) \;\; (18, 5) \;\; (19, 15)$$

Se expresa cada bloque como un número en base $K = 26$

$$(3, 15) = 3 \cdot 26^0 + 15 \cdot 26^1 = 393$$

$$(14, 7) = 14 \cdot 26^0 + 7 \cdot 26^1 = 196$$

$$(18, 5) = 18 \cdot 26^0 + 5 \cdot 26^1 = 148$$

$$(19, 15) = 19 \cdot 26^0 + 15 \cdot 26^1 = 409$$

Por lo tanto el cifrado es $c = m^e \; (mod \; N)$

$$c_1 = 393^{37} \; (mod \; 899) = 662$$

$$c_2 = 196^{37} \; (mod \; 899) = 299$$

$$c_3 = 148^{37} \ (mod\ 899) = 282$$
$$c_4 = 409^{37} \ (mod\ 899) = 688$$

Se tiene que el mensaje cifrado es: 662299282688

Para descifrar se usa $m = c^d \ (mod\ N)$

$$662^{613}\ mod\ 899 = 393$$
$$299^{613}\ mod\ 899 = 196$$
$$282^{613}\ mod\ 899 = 148$$
$$688^{613}\ mod\ 899 = 409$$

$$393 = 3 \cdot 26^0 + 15 \cdot 26^1 \rightarrow (3, 15) \rightarrow (co)$$
$$196 = 14 \cdot 26^0 + 7 \cdot 26^1 \rightarrow (14, 7) \rightarrow (ng)$$
$$148 = 18 \cdot 26^0 + 5 \cdot 26^1 \rightarrow (18, 5) \rightarrow (re)$$
$$409 = 19 \cdot 26^0 + 15 \cdot 26^1 \rightarrow (19, 15) \rightarrow (so)$$

■

Capítulo 6

Implementación de los algoritmos RSA y Hill en software

6.1. Implementación del algoritmo RSA

```
#include <cstdlib>
#include <iostream>
#include <string>
#include <cstdio>
#include <fstream>
#include <gmp.h>
#include <locale.h>
using namespace std;
using std::string;


string alfabeto("abcdefghijklmn opqrstuvwxyz      ,.");

bool es_primo(int n)
{
  for(int i=2;i<n;i++)
    if(n%i==0)
     return false;

   return true;
}


int alg_euc(int a,int b)
{
   int max,min,r;

  // identificamos el mayor y menor de los numeros
   if(a>=b)
   {max=a;min=b;}
   else
   {max=b;min=a;}

   while(min!=0)
```

```
    {
     r=max%min;
     max=min;
     min=r;
    }
     return max;
}


long* alg_euc_ext(int n1,int n2) // n1 es a y n2 es b
{
    long array[3],x=0,y=0,d=0,x2 = 1,x1 = 0,y2 = 0,y1 = 1,q = 0, r = 0;
   if(n2==0)
   {
   array[0]=n1;
   array[1]=1;
   array[2]=0;
   }
   else
   {
    while(n2>0)
       {
      q = (n1/n2);
       r = n1 - q*n2;
       x = x2-q*x1;
       y = y2 - q*y1;
       n1 = n2;
       n2 = r;
       x2 = x1;
       x1 = x;
       y2 = y1;
       y1 = y;
       }
       array[0] = n1;    // mcd (n1,n2)
       array[1] = x2;    // x
       array[2] = y2;    // y
     }
     return array;
}


long Inverso_Zn(int a,int n)
{
  long* ptr,array[3];
  ptr=alg_euc_ext(n,a);

  array[0]=*ptr;
  array[1]=*(ptr+1);
  array[2]=*(ptr+2);

  if(array[0]!=1)
  return -1;
  else
  {
  if(array[2]<0)
```

```
    array[2]+=n;
    return array[2];
   }
}


unsigned long long  Exponenciacion_Zn(unsigned long long  a,unsigned long
    long  k,unsigned long long  n)
{
  // convertimos "k" a binario
    unsigned long long numero=k;

  unsigned long long bin[500];
  unsigned long long  ind=0;
  while(numero>=2)
  {
   bin[ind++]=numero%2;
   numero/=2;
  }
  bin[ind]=numero;
  unsigned long long  tam=ind+1;

  unsigned long long  b=1;
  if(k==0)
     return b;

  unsigned long long  A=a;
  for(int i=(tam-1);i>=0;i--)
  {
    b=(b*b)%n;
    if(bin[i]==1)
    b=(A*b)%n;
   // cout<<"b :"<<b<<endl;
  }

return b;
}


int get_pos(string str,char elemento)
{
  for(int i=0;i<str.size();i++)
    if(str.at(i)==elemento)
      return i;
   return -1;
}


string  validar_mensaje(string texto_plano)
{
  string texto_plano_valido="";

// eliminamos los espacios del texto plano
  for(int i=0;i<texto_plano.size();i++)
     if(texto_plano.at(i)!=' ')
```

```
        texto_plano_valido+=texto_plano.at(i);

 // completamos con x al final para que sea potencia de 2
   int tam=texto_plano_valido.size();
   if(tam%2!=0)
    texto_plano_valido+="x";


  return texto_plano_valido;
}


int RSA()
{
     long int p,q,n,fi,e,d;
     string mensaje,mensaje_valido;
     char mensaje_aux[700];
       cout<<" ALGORITMO RSA :\n\n";
    // Debemos seguir una serie de pasos para generar las claves publica y
        privada :

    /* 1) Generamos aleatoriamente dos enteros p y q que deben de ser
        primos */
         do
         {p=rand()%500000+20000;
         }while(!es_primo(p));

         do
         {q=rand()%500000+20000;
         }while(!es_primo(q));

         cout<<" p : "<<p<<"\n q : "<<q;

    /* 2) Calculamos el valor de n */
         n = p * q;
         cout<<"\n n : "<<n;


    /* 3) Calculamos el valor de fi */
         fi = (p-1) * (q-1);
         cout<<"\n fi : "<<fi;

    /* 4) Seleccionamos aleatoriamente un entero 'e' tal que mcd(e,fi)=1 y
        1 < e < fi */
       do{
       e= rand()%(fi-2)+2;
       }while(alg_euc(e,fi)!=1);
       cout<<"\n e : "<<e;

    /* 5) Usar el algoritmo de euclides extendido para hallar un entero 'd'
         tal que
            ed = 1 (mod fi) donde 1 < d < fi (en otras palabras, hallar el
                inverso de 'e') */
         d = Inverso_Zn(e,fi);
         cout<<"\n d : "<<d;
```

```
    /*  6) La clave publica es (n,e) y la clave privada es d */
        cout<<"\n\n clave publica : ("<<n<<" , "<<e<<")";
        cout<<"\n clave privada : "<<d<<endl<<endl;

   ////////////////////////////////////////////////

    //cout<<"Mensaje a encriptar: ";
    string nombre;
   ////////////////////////////////
    const string nfichero = "mensajeEntrada.txt";
    char cadena[700];
    ifstream fichero;
    fichero.open(nfichero.c_str());
    if(!fichero.fail()){
      fichero.getline(cadena,700,'\n');
    while(!fichero.eof()){}

    mensaje=cadena;
      cout<<" \n mensaje de entrada : \n\n"<<mensaje<<endl;
    mensaje_valido=validar_mensaje(mensaje);
      cout<<" \n mensaje valido : \n\n"<<mensaje_valido<<endl;

    // representamos numericamente el mensaje
      cout<<" \n mensaje en numeros conforme al alfabeto : \n\n";
    long int mensaje_int[mensaje_valido.size()]; /*posiciones de los
        caracteres en el alfabeto del mensaje*/
    long int mensaje_cifrado[mensaje_valido.size()/2];

    for(int i=0;i<mensaje_valido.size();i++)
      mensaje_int[i]=get_pos(alfabeto,mensaje_valido.at(i));
    for(int i=0;i<mensaje_valido.size();i++)
      cout<<mensaje_int[i]<<" ";
    cout<<endl;
    cout<<" \n mensaje agrupado en pares de dos letras : \n\n";

    // agrupamos de 2 en 2 el mensaje numerico
     for(int i=0;i<(mensaje_valido.size()/2);i++)
       mensaje_cifrado[i]=mensaje_int[i*2]*100+mensaje_int[i*2+1];
    for(int i=0;i<(mensaje_valido.size()/2);i++)
       cout<<mensaje_cifrado[i]<<" ";
     cout<<endl;

     cout<<"\n mensaje cifrado : \n\n";
     // elevamos al cuadrado el mensaje_cifrado
     for(int i=0;i<(mensaje_valido.size()/2);i++)
       mensaje_cifrado[i]=Exponenciacion_Zn(mensaje_cifrado[i],e,n);
     for(int i=0;i<(mensaje_valido.size()/2);i++)
       cout<<mensaje_cifrado[i]<<" ";
     cout<<endl;


    cout<<" \n\n mensaje cifrado a la potencia 'd':\n\n";
     for(int i=0;i<(mensaje_valido.size()/2);i++)
          mensaje_cifrado[i]=Exponenciacion_Zn(mensaje_cifrado[i],d,n);
```

```
       for(int i=0;i<(mensaje_valido.size()/2);i++)
            cout<<mensaje_cifrado[i]<<" ";
       cout<<endl;

      cout<<"\n\n Se convierte el mensaje a numeros conforme el alfabeto \n\
          n";
      for(int i=0;i<(mensaje_valido.size()/2);i++)
        {mensaje_int[i*2]=mensaje_cifrado[i]/100;
         mensaje_int[i*2+1]=mensaje_cifrado[i]%100;
        }
      for(int i=0;i<((mensaje_valido.size()/2)*2);i++)
           cout<<mensaje_int[i]<<" ";
      cout<<endl;

string mensajeS;
      // hallamos el mensaje
      cout<<"\n\n mensaje de salida   : \n\n";
      for(int i=0;i<((mensaje_valido.size()/2)*2);i++)
        mensajeS+=alfabeto.at(mensaje_int[i]%34);
      cout<<mensajeS;

ofstream ficheroSalida;
 ficheroSalida.open ("mensajeSalida.txt");
 ficheroSalida << mensajeS;
 ficheroSalida.close();

 return 0;
}
}



int main(int argc, char *argv[])
{
     setlocale(LC_CTYPE, "Spanish");
     int op;
     cout<<"\n\n ALGORITMO RSA\n\n";

     cout<<"     [1] -> Empezar algoritmo\n\n";
     cout<<"     [2] -> Salir\n\n";
     cout<<" Seleccione una opcion:";
     cin>>op;
     if(op==1)
       RSA();
     if(op==2)
       EXIT_FAILURE;
     cout<<endl;

     system("PAUSE");
     return EXIT_SUCCESS;
}
```

6.2. Ejecución del algoritmo RSA

El programa genera aleatoriamente todo, a excepción del mensaje a encriptar, ya que este mensaje se escribe a partir de un archivo txt, llamado "mensajeEntrada.txt", y tiene como salida el mensaje desencriptado que se manda a otro archivo txt llamado "mensajeSalida.txt".

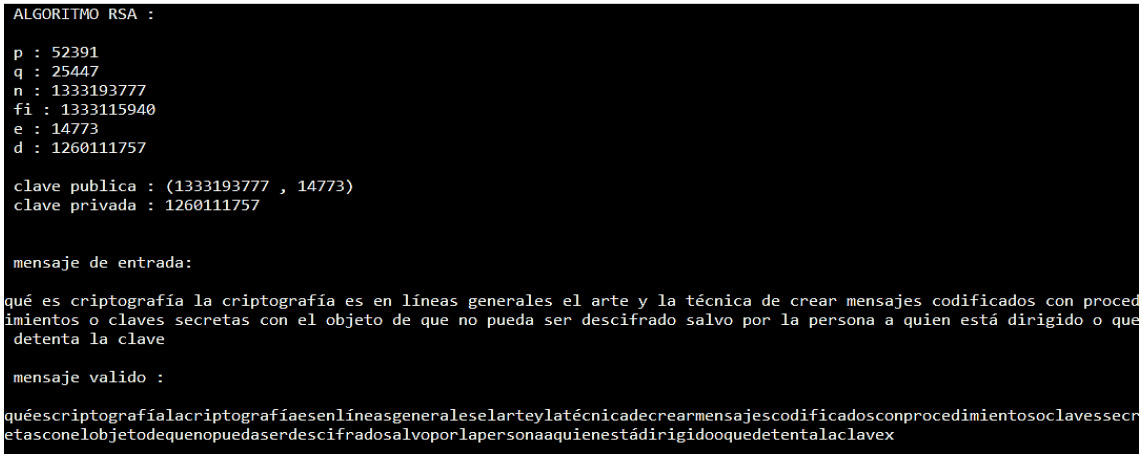


Figura 6.1: En esta figura se puede mostrar que todo el programa genera todo aleatoriamente, y se muestran las llaves públicas y privadas, así como también se observa el mensaje a encriptar proveniente del txt de entrada, como es que se le quitan los espacios a dicho mensaje para poder ser encriptado.

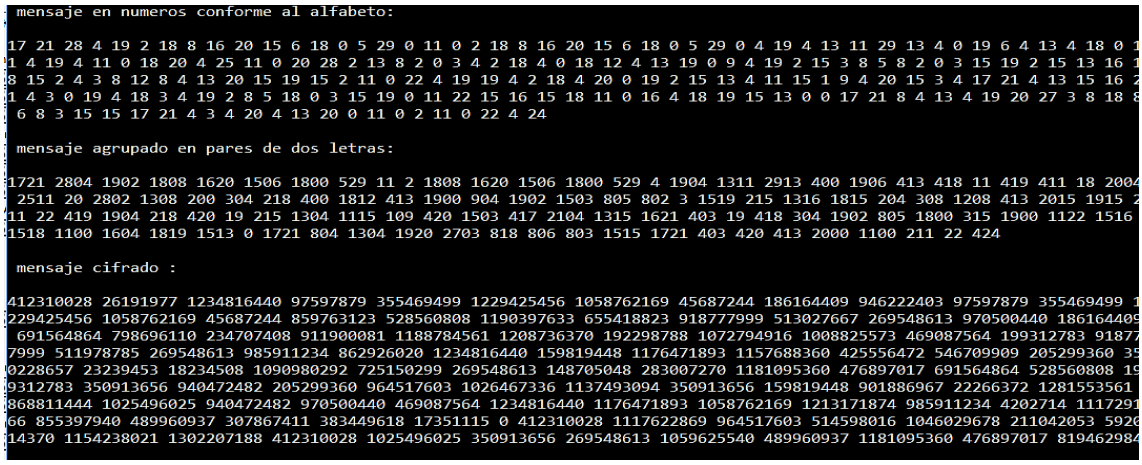


Figura 6.2: En esta figura se puede mostrar como es que el mensaje de entrada es convertido a su equivalente a número conforme va el alfabeto, para después poder aplicar el algoritmo RSA y que el mensaje quede encriptado como se puede observar en el apartado de la figura de mensaje cifrado.

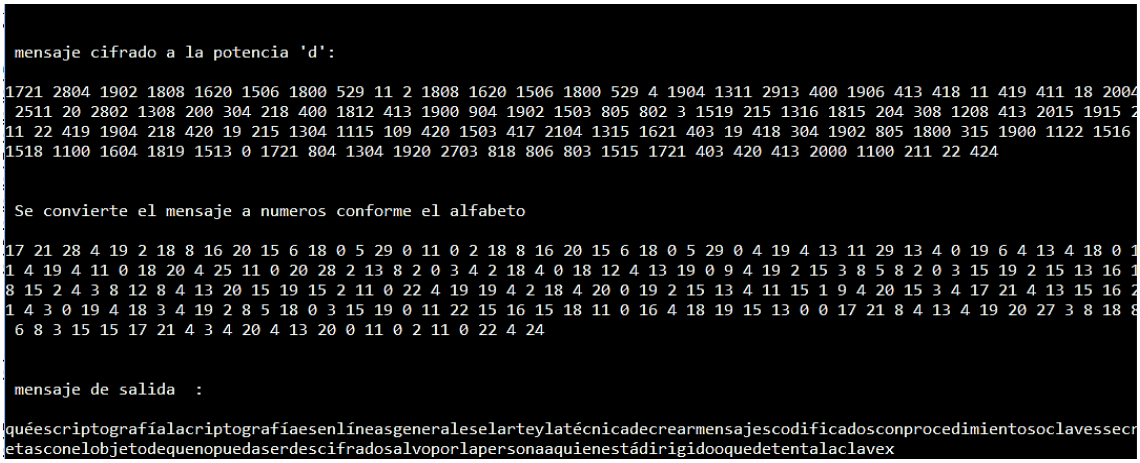


Figura 6.3: En esta figura se observa el proceso de descifrado, se observa como el mensaje cifrado es llevado a su equivalente númerico del alfabeto utilizado mediante el algoritmo RSA para después solo convertir cada número a letra y poder descifrar el mensaje de salida y este es mandado al txt correspondiente.

Se realizarón mas corridas para este programa con diferentes tipos de mensajes, los resultados se muestra a continuación en las siguientes figuras.

Corrida dos.

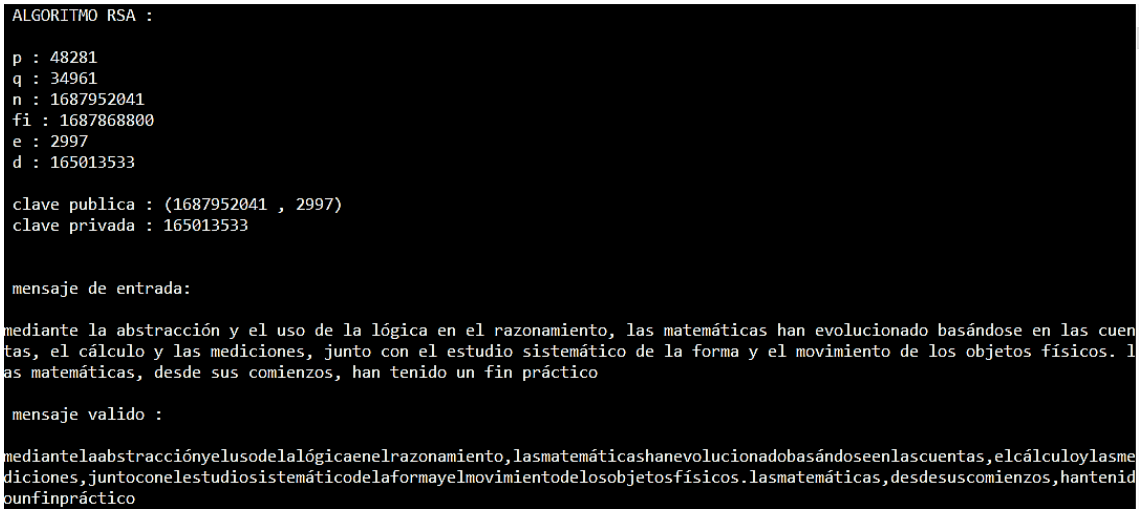


Figura 6.4

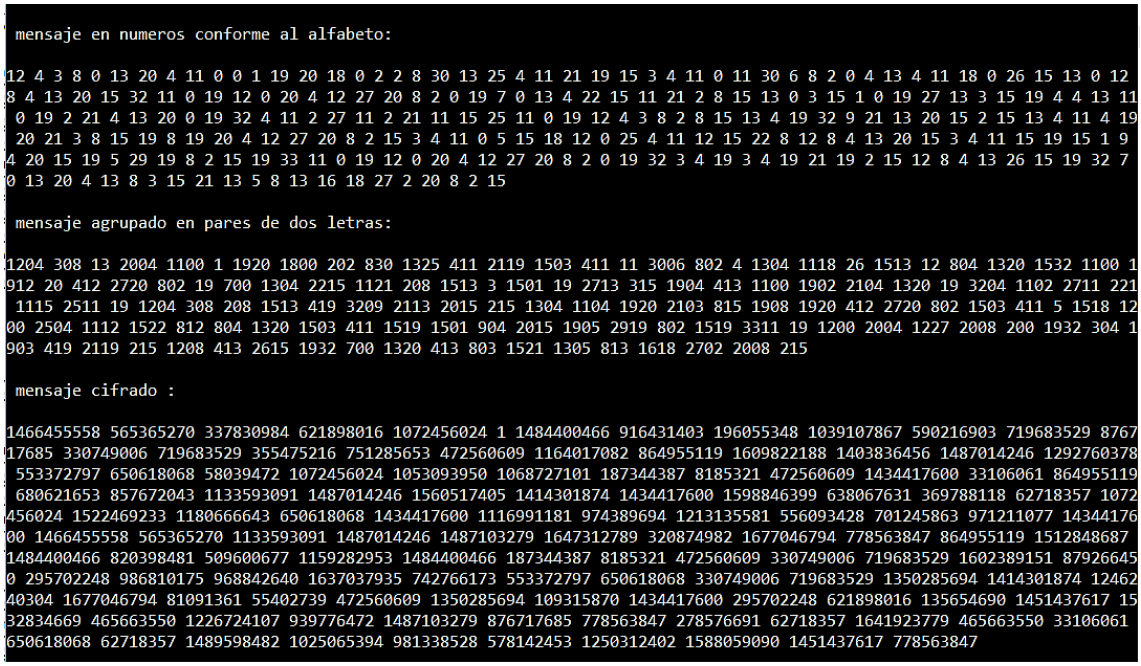


Figura 6.5

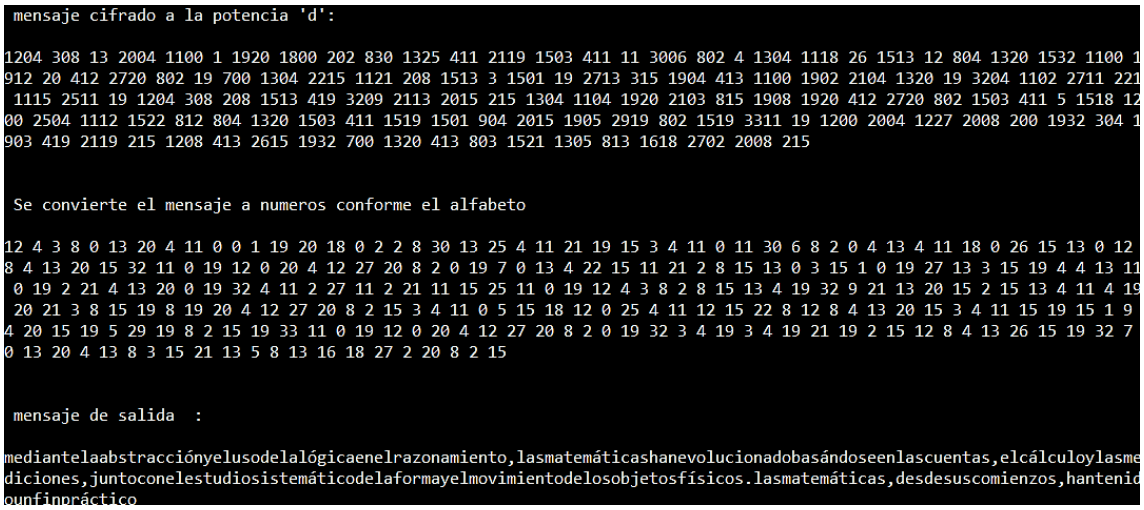


Figura 6.6

Corrida dos.

```
ALGORITMO RSA :

p : 42467
q : 33961
n : 1442221787
fi : 1442145360
e : 4829
d : 503511509

clave publica : (1442221787 , 4829)
clave privada : 503511509


mensaje de entrada:

el refrán es una oración breve, de carácter sentencioso y de fácil memorización, con muy poco que se diga el interlocuto
r entiende perfectamente que es lo que se trasmite, se enmarca dentro del lenguaje popular, es sencillo y de simplicidad
gráfica

mensaje valido :

elrefránesunaoraciónbreve,decaráctersentenciosoydefácilmemorización,conmuypocoquesedigaelinterlocutorentiendeperfectamen
tequeesloquesetrasmite,seenmarcadentrodellenguajepopular,essencilloydesimplicidadgráfica
```

Figura 6.7

```
mensaje en numeros conforme al alfabeto:

4 11 18 4 5 18 27 13 4 19 21 13 0 15 18 0 2 8 30 13 1 18 4 22 4 32 3 4 2 0 18 27 2 20 4 18 19 4 13 20 4 13 2 8 15 19 15
25 3 4 5 27 2 8 11 12 4 12 15 18 8 26 0 2 8 30 13 32 2 15 13 12 21 25 16 15 2 15 17 21 4 19 4 3 8 6 0 4 11 8 13 20 4 18
11 15 2 21 20 15 18 4 13 20 8 4 13 3 4 16 4 18 5 4 2 20 0 12 4 13 20 4 17 21 4 4 19 11 15 17 21 4 19 4 20 18 0 19 12 8 2
0 4 32 19 4 4 13 12 0 18 2 0 3 4 13 20 18 15 3 4 11 11 4 13 6 21 0 9 4 16 15 16 21 11 0 18 32 4 19 19 4 13 2 8 11 11 15
25 3 4 19 8 12 16 11 8 2 8 3 0 3 6 18 27 5 8 2 0

mensaje agrupado en pares de dos letras:

411 1804 518 2713 419 2113 15 1800 208 3013 118 422 432 304 200 1827 220 418 1904 1320 413 208 1519 1525 304 527 208 111
2 412 1518 826 2 830 1332 215 1312 2125 1615 215 1721 419 403 806 4 1108 1320 418 1115 221 2015 1804 1320 804 1303 416 4
18 504 220 12 413 2004 1721 404 1911 1517 2104 1904 2018 19 1208 2004 3219 404 1312 18 200 304 1320 1815 304 1111 413 62
1 9 416 1516 2111 18 3204 1919 413 208 1111 1525 304 1908 1216 1108 208 300 306 1827 508 200

mensaje cifrado :

1041921936 836347573 1359132529 68260344 588309183 452387434 674301935 666659617 65941697 759838598 661631981 1178415219
 349443498 300049746 1217590213 249323547 1016975953 582961066 535759604 206904710 220878659 65941697 1050240116 1430818
919 300049746 1247887151 65941697 782140321 243764910 399824505 735886485 52800659 1370838220 391254675 1415216156 33807
6487 661834590 689112291 1415216156 866650044 588309183 114138410 1370532817 1132147126 336284186 206904710 582961066 12
80405259 448135476 781295346 836347573 206904710 1290624318 1024248990 812321894 582961066 87688054 1016975953 703115919
 220878659 1184523780 866650044 538598009 773927218 253921867 673070082 535759604 1323983107 483107591 1321756461 118452
3780 1093329583 538598009 338076487 1256549103 1217590213 300049746 206904710 1269610551 300049746 1283281384 220878659
928685720 915921606 812321894 1049491099 817403181 1256549103 752767903 313349634 220878659 65941697 1283281384 14308189
19 300049746 351133844 834143623 336284186 65941697 1398576579 871263856 249323547 891684041 1217590213
```

Figura 6.8

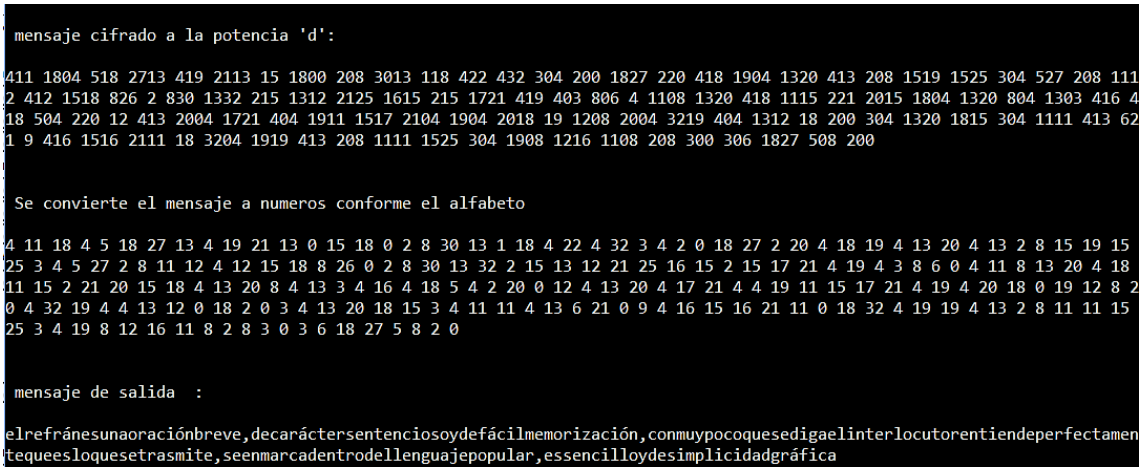

mensaje cifrado a la potencia 'd':
411 1804 518 2713 419 2113 15 1800 208 3013 118 422 432 304 200 1827 220 418 1904 1320 413 208 1519 1525 304 527 208 111
2 412 1518 826 2 830 1332 215 1312 2125 1615 215 1721 419 403 806 4 1108 1320 418 1115 221 2015 1804 1320 804 1303 416 4
18 504 220 12 413 2004 1721 404 1911 1517 2104 1904 2018 19 1208 2004 3219 404 1312 18 200 304 1320 1815 304 1111 413 62
1 9 416 1516 2111 18 3204 1919 413 208 1111 1525 304 1908 1216 1108 208 300 306 1827 508 200
Se convierte el mensaje a numeros conforme el alfabeto
4 11 18 4 5 18 27 13 4 19 21 13 0 15 18 0 2 8 30 13 1 18 4 22 4 32 3 4 2 0 18 27 2 20 4 18 19 4 13 20 4 13 2 8 15 19 15
25 3 4 5 27 2 8 11 12 4 12 15 18 8 26 0 2 8 30 13 32 2 15 13 12 21 25 16 15 2 15 17 21 4 19 4 3 8 6 0 4 11 8 13 20 4 18
11 15 2 21 20 15 18 4 13 20 8 4 13 3 4 16 4 18 5 4 2 20 0 12 4 13 20 4 17 21 4 4 19 11 15 17 21 4 19 4 20 18 0 19 12 8 2
0 4 32 19 4 4 13 12 0 18 2 0 3 4 13 20 18 15 3 4 11 11 4 13 6 21 0 9 4 16 15 16 21 11 0 18 32 4 19 19 4 13 2 8 11 11 15
25 3 4 19 8 12 16 11 8 2 8 3 0 3 6 18 27 5 8 2 0
mensaje de salida :
elrefránesunaoraciónbreve,decaráctersentenciosoydefácilmemorización,conmuypocoquesedigaelinterlocutorentiendeperfectamen
tequeesloquesetrasmite,seenmarcadentrodellenguajepopular,essencilloydesimplicidadgráfica

Figura 6.9

6.3. Implementación del algoritmo Hill

```
#include <conio.h>
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <sstream>
#include <locale.h>
#include <fstream>

using namespace std;


int determinanteMatriz(int M[][100],int d);
char *agrupar(char *text,char *textAgru,int d);



int *euclidesExtendido(int a, int b){

 int d,x,y;
 int x1,x2,y1,y2;
 int q,r;
 int *rpts=new int[3];

 if(b==0){
  d=a;
  x=1;
  y=0;

  rpts[0]=d; rpts[1]=x; rpts[2]=y;
  return rpts;
 }
 x1=0; x2=1;
 y1=1; y2=0;
 while(b >0){
  q=(a/b); r=a-q*b;
  x=x2-q*x1; y=y2-q*y1;
  a=b; b=r;
  x2=x1; x1=x;
  y2=y1; y1=y;
 }
 rpts[0]=a; rpts[1]=x2; rpts[2]=y2;

 return rpts;
}

int inversoMultiplicativo(int a, int z){

 int *resp;
 int inver;

 resp=euclidesExtendido(a,z);
```

```
 if(resp[0]==1){
  if(resp[1]< 0)
   inver=z+resp[1];
  else if(resp[1] >0)
   inver=resp[1];
  delete(resp);
  return inver;
 }
 else {
  delete(resp);
  return -1;
 }

}


void generarMatriz(int d,int M[][100]){


 for(int j=0;j<d;j++)
        for(int i=0;i<d;i++)
        M[i][j]=rand()%25;

   cout<<"\n La matriz aleatoria es: \n"<<endl;

   for(int j=0;j<d;j++)
        {for(int i=0;i<d;i++)
         cout<<M[i][j]<<" \t";
       cout<<endl;
  }

}


void multiplicarMatriz(int M[][100],int P[][100], int C[][100], int m,int p
    ,int n){
 int s;

 for (int i=0;i< m;i++){
  for(int j=0;j< n;j++){
   s=0;
   for(int k=0;k< p;k++){
    s+=M[i][k]*P[k][j];
   }
   C[i][j]=s;
  }
 }
}

int subMatriz(int i, int j, int M[][100],int temp[][100],int d){

 int fil=0;
 int col=0;
```

```
 for(int k=0;k< d;k++){
  if(k!=i){
   col=0;
   for(int l=0;l< d;l++){
    if(l!=j){
     temp[fil][col]=M[k][l];
     col++;
    }

   }
   fil++;
  }

 }
 return determinanteMatriz(temp,d-1);
}


int determinanteMatriz(int M[][100],int d){

 int temp[100][100];

 if(d==2){
  int deter=M[0][0]*M[1][1]-M[1][0]*M[0][1];

  return deter;
 }else{
  int deter=0;

  for(int j=0;j< d;j++){
   subMatriz(0,j,M,temp,d);
   deter=deter+pow(-1,0+j)*M[0][j]*determinanteMatriz(temp,d-1);

  }

  return deter;
 }

}

bool comparar(char *clave,char a,int k){

 for(int i=0;i< k;i++){
  if(clave[i]==a)
   return false;
 }

 return true;
}

char *preMatriz(char *clave,char *letras,char *textCpy){
 //alfabeto desde codigo ascii
 int k=0,i=0;
 bool band=true;
```

```
 while(k< 25){

  if(clave[k]=='j')
   clave[k]='i';

  if(clave[k]!='\0' && band==true){
   if(clave[k]!='_' && clave[k] >96 && clave[k]< 123){
       //  if(clave[k]!=' ' && clave[k] >96 && clave[k]< 123){
    if(comparar(clave,clave[k],k)){
     textCpy[i]=clave[k];
     i++;
    }
   }
  }
  else{
   if(band==true)
    k=0;
   band=false;
   if(comparar(textCpy,letras[k],i)){
    textCpy[i]=letras[k];
    i++;
   }
  }

  k++;
 }
 textCpy[25]='\0';
 return textCpy;
}

int algoCifradoHill( int M[][100], int d){

      string nombre;
      string mensaje;
     ////////////////////////////
      const string nfichero = "mensajeEntrada.txt";
      char text[700];
      ifstream fichero;
      fichero.open(nfichero.c_str());
      if(!fichero.fail()){
      fichero.getline(text,700,'\n');
      while(!fichero.eof()){}

 cout<<"\n\n"<<text;
 char textAgru[500];
 char texEnc[500];
 char textDes[500];
 int P[100][100],p=0;
 int C[100][100];
 int inver,deter,verfInver;
 int k=0,m=0,ini=0,tam;

 deter=determinanteMatriz(M,d)%26;

 if(deter< 0){
```

```
  verfInver=deter+26;
  cout<< "\n\n Determinante: "<< verfInver<< "\n";
 }
 else{
  verfInver=deter;
    cout<< "\n\n Determinante: "<< verfInver<  "\n";
 }

 inver=inversoMultiplicativo(verfInver,26);
 cout<< "\n\n inverso("<< verfInver<< ",26) = "<< inver<< "\n";

 if(deter!=0 && inver!=-1){
  strcpy(textAgru,agrupar(text,textAgru,d));

  tam=strlen(textAgru);

  cout<< "\n Texto: "<< textAgru;

  while(k<=tam){

   if(p< d){
    P[p][0]=textAgru[k]-97;

    p++;
   }else{
    multiplicarMatriz(M,P,C,d,p,1);
    for(int i=ini;i< k;i++){
     textAgru[i]=(C[m][0]%26)+97;
     m++;

    }
    ini=k+1;
    m=0;
    p=0;

   }
   k++;
  }

  cout << "\n\n Texto Encriptado: "<< textAgru;

  strcpy(textDes,textAgru);
 }else
  cout<< "\n\n Con la clave matrices generada no se puede encriptar el
      mensaje";

////////////////////////////////
/////desencriptamiento////////////


  int ini1=0,p1=0,k1=0,m1=0;
  int resul;
  int Q[100][100];
  int D[100][100];
  int A[100][100];
```

```
  int temp[100][100];


if(deter< 0){
  verfInver=deter+26;
  cout<< "\n\n Determinante: "<< verfInver<< "\n";
 }
 else{
  verfInver=deter;
  cout<< "\n\n Determinante: "<< verfInver<< "\n";
 }
 inver=inversoMultiplicativo(verfInver,26);
 cout<< "\n\n inverso("<< verfInver<< ",26) = "<< inver<< "\n";

 if(deter!=0 && inver!=-1){

  for(int i = 0; i <  d; i++)
   for(int j = 0; j <  d ;j++)
    A[j][i] = pow(-1,i+j)*subMatriz(i,j,M,temp,d);

  for(int i = 0; i <  d; i++){
   for(int j = 0; j <  d ;j++){
    resul=A[i][j]*inver;
    if(resul< 0)
     A[i][j]=(resul%26)+26;
    else
     A[i][j]=resul%26;
   }

  }
  int tam1=strlen(textDes);

  cout<< "\n\n Texto: "<< textDes;

  while(k1<=tam1){

   if(p1< d){
    Q[p1][0]=textDes[k1]-97;
    //cout< < P[p][0]< < " " ;
    p1++;
   }else{
    multiplicarMatriz(A,Q,D,d,p1,1);
    for(int i=ini1;i< k1;i++){
     textDes[i]=(D[m1][0]%26)+97;
     m1++;

    }
    ini1=k1+1;
    m1=0;
    p1=0;

   }
   k1++;
  }
```

```
 cout<< "\n\n Texto Desencriptado: "<< textDes;

 }else
  {cout<< "\n\n Con la clave matrices generada no se puede desencriptar el
      mensaje";}
 cout<< endl;

 FILE *archivo = fopen("mensajeSalida.txt","w");
 fprintf(archivo, textDes);
    fclose(archivo);

 return 0;

}
}


char *agrupar(char *text,char *textAgru,int d){

 int k=0,i=0;
 int cont=0;


 while(text[k]!='\0'){

  if(text[k]!=' '){
   if(cont!=d){
    textAgru[i]=text[k];
    cont++;
    i++;
   }
   else{
    textAgru[i]=' ';
    textAgru[i+1]=text[k];
    i+=2;
    cont=1;
   }

  }
  k++;
 }

 while(cont< d){
  textAgru[i]='x';
  cont++;
  i++;
 }
 textAgru[i]='\0';

 return textAgru;
}

int main(int argc, char *argv[]) {
    setlocale(LC_CTYPE, "Spanish");
 char text[500];
```

```
 int M[100][100];
 int d;
 int opcion;
 while(1)
 {
  system("cls");
  cout<< "\n\t\t\tALGORITMO DE ENCRIPTACION HILL\n";
  cout<< "1. INICIAR ALGORITMO\n";
  cout<< "2. SALIR\n";
  cout<< "INGRESE OPCION: ";
  cin >>opcion;
  switch(opcion)
  {
   case 1:
   {
    system("cls");
    cout<< "\n\t\t\tALGORITMOS DE ENCRIPTACION HILL\n";

    d=3+rand()%(6-3);
    generarMatriz(d,M);
    algoCifradoHill(M,d);
    cout<< endl;

    system("pause");
    break;
   }
   case 2:
   {

     EXIT_FAILURE;
    cout<< endl;
     system("PAUSE");
    return EXIT_SUCCESS;
    //system("pause");
    break;
   }
  }
 }
 return 0;
}
```

6.4. Ejecución del algoritmo Hill

El programa genera aleatoriamente las matrices que se necesitan, estas pueden ser de orden tres, cuatro, cinco y seis, a excepción del mensaje a encriptar, ya que este mensaje se escribe a partir de un archivo txt, llamado "mensajeEntrada.txt", y tiene como salida el mensaje desencriptado que se manda a otro archivo txt llamado "mensajeSalida.txt".

En la figura **6.10** se puede observar como es que se genere una matriz aleatoriamente, como es que se rcupera el mensaje a partir del txt de entrada, como es que se agrupa dicho mensaje en vectotes dependiendo del tamaño de la matriz y despues se encripta el mensaje con el algoritmo Hill.

En la figura **6.11** se puede ver como se calcula el determinante y se calcula el inverso de la matriz generada en el programa. También se ve como es que se recupera el mensaje cifrado en paquetes del tamaño de la matriz para proceder a desencriptar con el algoritmo Hill y finalmente se desencripta el mensaje de salida en bloques para mandarlo a un txt de salida.

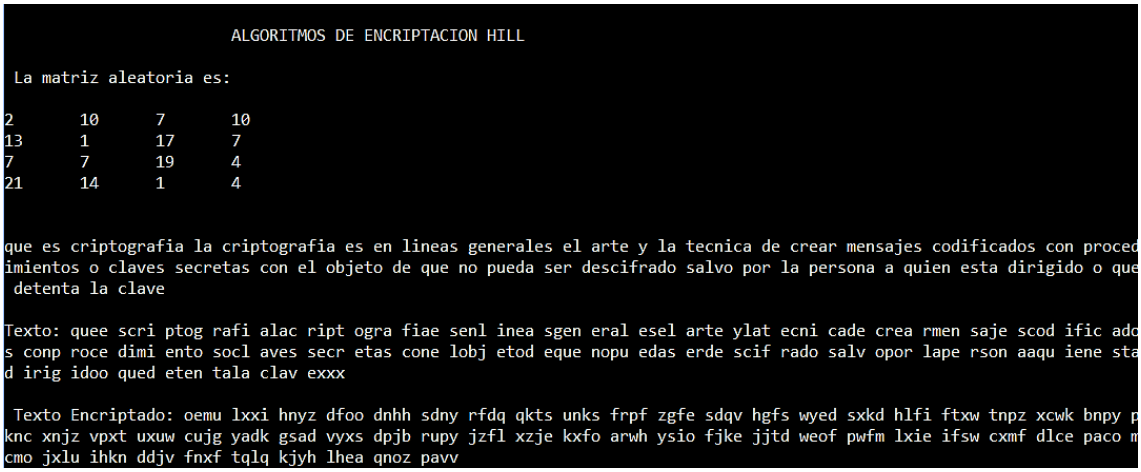


Figura 6.10

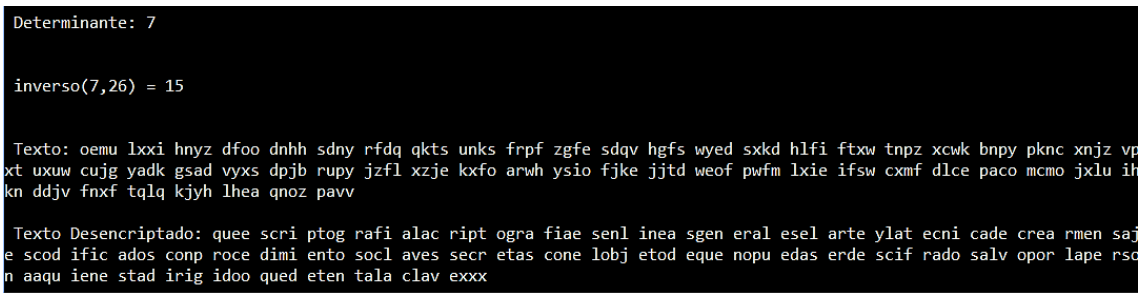


Figura 6.11

Se realiarón varias corridas con el programa a fin de ver que este no fallara, y se muestras a continuación los resultados de este.

Corrida dos

```
                    ALGORITMOS DE ENCRIPTACION HILL

 La matriz aleatoria es:

8       19      14      1       23
12      13      18      7       4
16      8       15      14      8
4       5       2       6       23
11      21      20      24      22


mediante la abstraccion y el uso de la logica en el razonamiento las matematicas han evolucionado basandose en las cuent
as el calculo y las mediciones junto con el estudio sistematico de la forma y el movimiento de los objetos fisicos las m
atematicas desde sus comienzos han tenido un fin practico

Texto: media ntela abstr accio nyelu sodel alogi caene lrazo namie ntola smate matic ashan evolu ciona dobas andos eenla
 scuen tasel calcu loyla smedi cione sjunt ocone lestu diosi stema ticod elafo rmaye lmovi mient odelo sobje tosfi sicos
 lasma temat icasd esdes uscom ienzo shant enido unfin pract icoxx

 Texto Encriptado: qgpag yjiik rlqbx imykw saedc dtxew alihc ulmyx arksi izqnr clcsm sboaz mupkg drnqy saajt yhsuv qbtbj
 qrrzo ubhux xpyyy fcknf ekzyw ioira klsel qnumf brclb ohcix skybq klujx qfcts fqizp asyfn ehkdn ewclg xwyir ukozr sltut
 owwts gaiuk ixgxx dyovf ltkuw qenea yqsaq czhcr vnedp ouupn zshvi vfqet hegyn
```

Figura 6.12

```
 Determinante: 25


 inverso(25,26) = 25


 Texto: qgpag yjiik rlqbx imykw saedc dtxew alihc ulmyx arksi izqnr clcsm sboaz mupkg drnqy saajt yhsuv qbtbj qrrzo ubhu
x xpyyy fcknf ekzyw ioira klsel qnumf brclb ohcix skybq klujx qfcts fqizp asyfn ehkdn ewclg xwyir ukozr sltut owwts gaiu
k ixgxx dyovf ltkuw qenea yqsaq czhcr vnedp ouupn zshvi vfqet hegyn

 Texto Desencriptado: media ntela abstr accio nyelu sodel alogi caene lrazo namie ntola smate matic ashan evolu ciona do
bas andos eenla scuen tasel calcu loyla smedi cione sjunt ocone lestu diosi stema ticod elafo rmaye lmovi mient odelo so
bje tosfi sicos lasma temat icasd esdes uscom ienzo shant enido unfin pract icoxx
```

Figura 6.13

Corrida tres

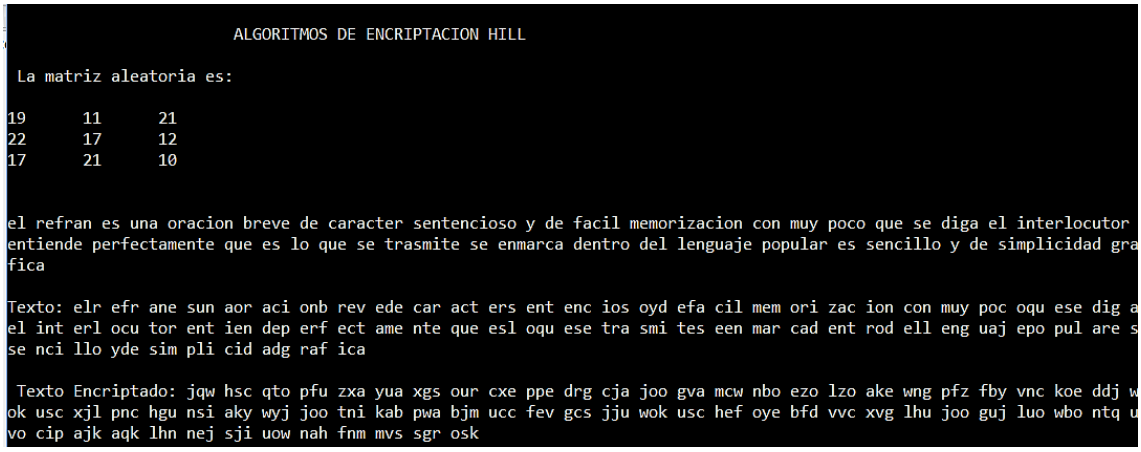


Figura 6.14

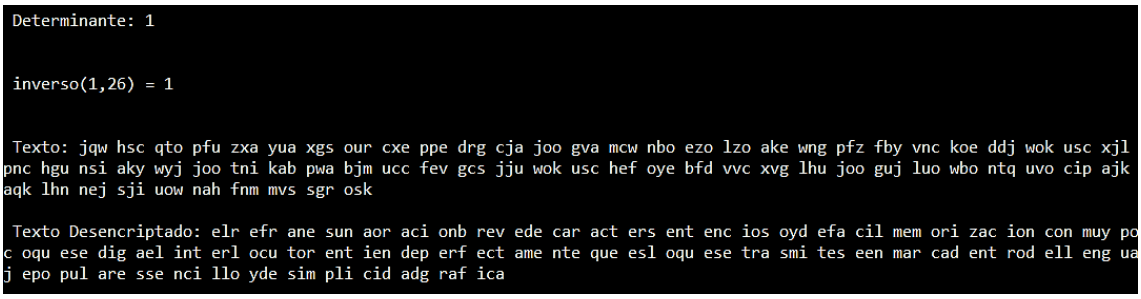


Figura 6.15

Conclusión

Como se pudo apreciar en el presente proyecto se implementaron dos algoritmos de encriptación el RSA y el Hill. Cada algoritmo tiene sus ventajas y desventajas, por ejemplo, el RSA una de sus principales ventajas es que se genera la llave privada y solo el emisor y receptor conocen esta, así que es difícil poder conseguirla para desencriptar el mensaje, pero siempre hay maneras de poder conseguir la llave privada, pero en general es un buen algoritmo siempre y cuando se usen números primos gigantes, es decir, con decenas de dígitos, pero esto depende de la cantidad de memoria que tenga la maquina donde se están generando estos números primos, ya que se puede atacar mediante dos números primos y así conseguir el modulo conseguido, hasta poder encontrar los dos números primos. Hablando del algoritmo Hill, también es un buen algoritmo, que necesita de una matriz que cumpla ciertas propiedades lo cual lo hace difícil descifrar, pero a la vez es un poco difícil encontrar la matriz que cumpla las propiedades, este algoritmo muy atacado ya que como es lineal es facial hacer un ataque con un texto plano conocido con el método de Gauss Jordan y encontrar así la matriz clave k. En conclusión puedo decir que se cumplieron los objetivos del proyecto y se pudo ver las ventajas y desventajas de cada algoritmo, en mi opinión el algoritmo mas fuerte y difícil de desencriptar es el RSA.

Bibliografía

[1] A. Álvarez Gaona, *"Implementación en software-hardware de aritmética sobre campos finitos binarios F_{2m} en curva elípticas para aplicaciones criptográficas de llave pública"*, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónogma Metropolitana Azcapotzalco, México, 2012.

[2] A. Haraty and Hadi Otrok, *"Attacking ElGamal based cryptographic algorithms using Pollards rho algorithm"*, in The 3rd ACS/IEEE International Conference on, 2005.

[3] Amparo Fúster, Dolores de la Guía Martínez, Luis Hernández Encinas, Fausto Montoya Vitini y Jaime Muñoz Masqué, *"Técnicas Criptográficas de protección de datos"*, 2a ed. Alfaomega, 2001.

[4] Amparo Fúster, Luis Hernández Encinas, Agustín Martín Muõz, *"Criptograifa, proyección de datos y aplicaciones: una guía para el estudiante"*, 3a ed. Alfaomega, 2012.

[5] B. Ovilla Martínez, *"Codiseño hardware-software de algoritmo de cifrado DES"*, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007.

[6] David R. Kohel "Cryptography", Creative Commons Attribution-Noncommercial-Share Alike 3.0 Unported License, 2008.

[7] Del Valle Sotelo Juan Carlos, *"Álgebra lineal para estudientes de ingeniería y ciencias"*, a ed. Mc Graw Hill, México D,F, 2011.

[8] G. Granados Paredes, *"Introducción a la Criptografía"*, Revista Digital Universitaria, vol. 7, no. 7, pp. 7-12, 2006.

[9] Howard Anton *"Algebra lineal"*, 3a edición, ed Limusa, México D.F, 1994.

[10] J. M. R. Castillo Alamilla, "Desarrollo de mecanismos de seguridad de información en redes de datos para la creación de un sitio de comercio electrónico", Facultad de Ingeniería UNAM, México, 2008.

[11] Jeffrey Hoffstein, Jill Pipher and Joseph H. Silverman, *"An Introduction to Mathematical Cryptography"*, 1st ed. Springer, 2008, pp. 59-176.

[12] María Soledad Villar Lozano, *"Criptografía en curvas de Pell y generalizaciones"*, facultad de ciencias, Universidad de la República de Uruguay, Uruguay, 2010.

[13] R. Vargas Márquez, *"Implementación del algoritmo de cifrado AES mediante la metodología Hardware-Software"*, proyecto terminal, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010.

[14] Ramio Aguirre Jorge, *"Libro Electrónico de Seguridad Informática y Criptografía"*, Versión 4.1, pp. 40-48 , http://www.criptored.upm.es/guiateoria/gtm001a.html,2006.

[15] Ramzi A. Haraty, A. El-Kassar and Hadi Otrok, *"A comparitive Study of RSA based Cryptographic Algorithms"*, 13th International Conference on Intelligent and Adaptive Systems and Software Engineering, France, 2004.