Universidad Nacional Autónoma de México.
Facultad de Contaduría y Administración

Fundamentos matemáticos para el criptoanálisis de un algoritmo de sustitución polialfabética.

TESIS PROFESIONAL QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN INFORMÁTICA

PRESENTA

Jorge Alejandro Carrillo Ugalde.

Asesor: *M. en C. Leobardo Hernández Audelo.*

México, D.F. 20000

Reconocimientos

Para la elaboración del presente trabajo agradecemos a Sandra Díaz Santiago y Julia I. Muñoz Guerra del Departamento de matemáticas de la Universidad Autónoma Metropolitana campus Iztapalapa, porque con su entusiasmo y dedicación fue posible desarrollar algunos trabajos, los cuales sirvieron como motivación para escribir la presente Tesis.

Tanto en la corrección de estilo, ortografía y redacción, como en una infinidad de valiosos comentarios, agradecemos el esfuerzo interminable del Ing. Roque Alarcón Guerrero de la Coordinación De Información y Comunicación de la Facultad de Ingeniería.

Por último, sin ser menos importante, a la Dirección General de Servicios de Cómputo Académico de nuestra máxima casa de estudios en sus departamentos: Supercómputo, a cargo de Enrique Cruz Martínez, y el Laboratorio de Visualización, a cargo de José Luis Villareal, por proporcionarnos acceso a la Supercomputadora Origin 2000 y a diversos sistemas de cómputo que fueron necesarios para desarrollar diversos programas de cómputo, así como para utilizar diversas herramientas para la elaboración de diversas gráficas y figuras. También agradecemos sus valiosos consejos, paciencia y apoyo incondicional para la elaboración del presente trabajo.

Resumen

La necesidad de comunicarnos en nuestra vida se presenta a diario: escuchamos las noticias, mandamos cartas a los amigos, solicitamos ayuda y compartimos nuestras confidencias, por mencionar algunos ejemplos. Sin embargo, algunos mensajes sólo queremos comunicarlos con una persona o varias bien identificadas, por lo que requerimos de técnicas que nos auxilien a conseguirlo.

Al igual que para nosotros, esta necesidad de comunicación también se ve reflejada en muchas organizaciones. Si pensamos en alguna información estratégica, los mensajes que se intercambien referente a este tema deberán hacerse con cuidado, ya que al interior o exterior de una organización el mal uso de ella puede significar grandes pérdidas.

En un principio, cuando los documentos eran impresos en papel, éstos se guardaban en cajas fuertes custodiadas por feroces perros o se rodeaban por misiles nucleares en casos muy sofisticados y modernos. Sin embargo, con el uso de los sistemas de cómputo, ahora los datos se encuentran almacenados en forma de bits y la manera de protegerlos no es similar a la de un documento impreso.

Aunado a este problema, se encuentra el uso extensivo de redes de comunicación, donde fluyen grandes cantidades de datos que a su vez requieren ser protegidos durante su trayecto a través del medio de transmisión por el que viajan.

Un mecanismo para lograrlo es por medio de la *criptografía*, que desarrolla métodos y técnicas para ocultar la información. Este campo de estudio se convierte en un arma poderosa, obteniéndose con su aplicación diversos beneficios, entre los que se encuentran: verificar la identidad de las partes involucradas en una comunicación, lograr que un escrito sea ininteligible a los ojos de las personas no autorizadas y comprobar que un mensaje no haya sido modificado durante el trayecto por algún medio de comunicación, entre otros más.

El uso actual de la criptografía se ha extendido a casi todos los ámbitos donde la información y las comunicaciones juegan un papel crucial. Por estas razones es de vital importancia conocer no sólo la forma en la que opera, sino también las distintas técnicas que se utilizan para atacarla. Esto último es lo que se conoce como *criptoanálisis*, cuyo objetivo es conocer el significado de un mensaje secreto sin tener autorización para ello. Ambas disciplinas conforman lo que hoy conocemos como *criptología*.

Las técnicas actuales para ocultar la información requieren de una gran variedad de ideas matemáticas que necesitan de un alto nivel de abstracción y por consiguiente el criptoanálisis también.

Afortunadamente, las primeras técnicas criptográficas que utilizó la humanidad han contemplado una serie de conceptos fascinantes que se encuentran al alcance de la mayoría de las personas no especializadas en matemáticas. Cuando se aplica el criptoanálisis a éstas se adquieren los conocimientos fundamentales que nos ayudan a comprender las técnicas criptográficas actuales, así como también adquirimos una serie de razonamientos matemáticos interesantes que nos hacen reflexionar sobre su importancia en el mundo moderno y particularmente en la criptología.

Por otra parte, sabemos que es necesario fortalecer los vínculos entre los centros de investigación y las compañías de bienes y servicios con el fin de impulsar el desarrollo de nuestro país. En el caso de la criptografía, convergen un sinnúmero de líneas de investigación en matemáticas y computación consideradas como no aplicadas.

Por estas razones sería bueno que el Licenciado en Informática pudiera interactuar con grupos altamente especializados en dichas áreas, esfuerzo que se vería reflejado con el desarrollo de aplicaciones que satisfacerían las necesidades demandadas por la sociedad para proteger su información.

Por eso, además de contar con los conocimientos teóricos y técnicos en el área de computación y administración, también es necesario contar con una buena preparación en cuanto a matemáticas se refiere.

En el presente trabajo, estudiaremos a la criptografía desde sus entrañas con el objetivo de obtener un panorama sólido para adentrarnos en un camino cuya finalidad es la de forjar una mentalidad con la capacidad de enfrentar los nuevos retos en la protección de la información dentro de un mundo con vertiginosos cambios.

Dentro de la gran variedad de cursos de matemáticas a los que hemos asistido, es posible que se nos haya desarrollado un cierto temor. debido a que en ocasiones "no entendemos" las ideas que se presentan y ésto se ve reflejado en un bajo rendimiento en las evaluaciones académicas, incluso a las personas que se dedican a las matemáticas las identificamos con el estereotipo del "científico loco" de cualquier película, cuyos pensamientos son incomprensibles e inalcanzables.

La realidad está muy lejos de ésta visión y por eso abarcaremos de una forma natural e intuitiva una serie de conceptos matemáticos haciendo uso de los conocimientos que en la *Licenciatura en Informática* hemos adquirido, de tal suerte que seamos capaces de comprender los pilares principales en los que se basa la criptología.

Las matemáticas necesarias para lograrlo se encuentran relacionadas con: Teoría de la Información, a la que gracias a su estudio estableceremos una serie de resultados de gran interés para el criptoanálisis; Teoría de Números, donde investigaremos algunas propiedades relacionadas con los números enteros y, finalmente, con algunas estructuras algebraicas, que se estudian a detalle en el campo de las matemáticas llamado Álgebra.

Con lo anterior, uno de nuestros objetivos será el de comprender los fundamentos matemáticos básicos involucrados en la criptografía moderna. Nos introduciremos en este sendero realizando el criptoanálisis a un algoritmo de sustitución polialfabética conocido por el nombre de *Cifrado de Vigenére.*

Los cifrados de sustitución polialfabética basan su funcionamiento en sustituciones múltiples de caracteres. Este tipo de cifrados aparecieron por primer vez en 1568; año en el que se da a conocer el disco cifrador de Leon Battista Alberti. El *cifrado de Vigenére* es un caso particular, por lo que explicaremos su filosofía y justificaremos, mediante el criptoanálisis, las razones por las cuales ya no es utilizado en la actualidad.

Los trabajos previos que profundizan el criptoanálisis al cifrado de Vigenére [SIN66], son de gran interés para nuestro trabajo, ya que pretenderemos extender los resultados obtenidos con la finalidad de desarrollar un sistema de cómputo que realice lo que un criptoanalista debe hacer.

Con todo ésto, esperamos que el presente trabajo ilustre una manera de realizar criptoanálisis usando adecuadamente las matemáticas y, además, que muestre la potencialidad de la computación y las matemáticas en la criptología.

Índice General

Capítulo 1

Introducción

1.1 Criptología

Etimológicamente criptología significa *escritura en secreto.* Actualmente se le relaciona con la disciplina que desarrolla sistemas secretos y cuyas ramificaciones son: La *criptografía*, que tiene por objetivo diseñar sistemas secretos, y el *criptoanálisis*, que busca encontrar las fallas de éstos (Figura 1.1).

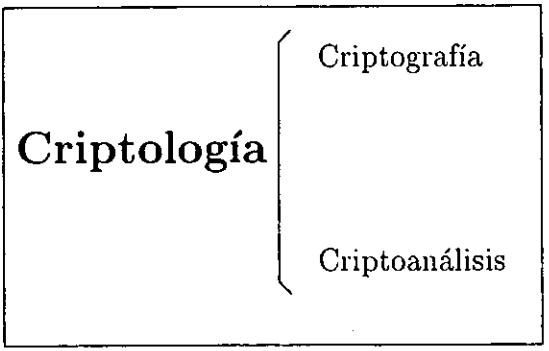


Figura 1.1: Criptología

En la criptografía, el diseño de sistemas secretos se concreta en la elaboración de un *criptosistema*, en la sección 1.2 aclararemos en lo que consiste, pero para comprenderlo es importante introducir una serie de términos de vital importancia.

Una vez comprendidos los tecnicismos básicos procederemos a mencionar el enfoque que utilizaremos en el estudio de la criptografía. así como los diversos modos en los que opera un criptoanalista. Esperamos entonces que se adquiera un panorama de las expresiones que en un futuro profundizaremos.

Para comenzar, el mensaje que será convertido en su forma secreta por medio de un algoritmo, que llamaremos *algoritmo de cifrado* E. se le conoce como *texto claro* M. El algoritmo de cifrado utiliza el texto M y una llave k para definir una función que transforma el texto original en otro indescifrable, que nombraremos *texto cifrado* C (Figura 1.2).

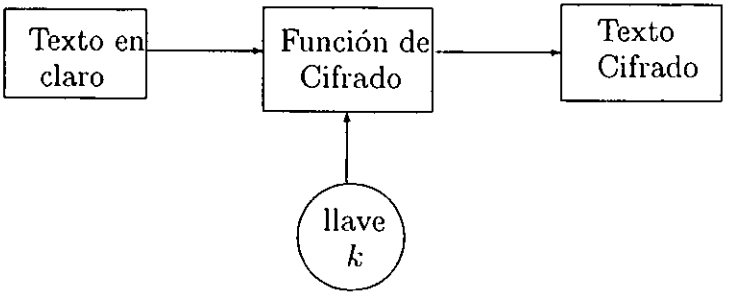


Figura 1.2: Cifrado de un Mensaje

Al agrupar todas las llaves k posibles, obtenemos un conjunto que lleva el nombre de *espacio de llaves* $\mathcal{K}$. Del mismo modo, al contemplar todos los textos en claro tenemos el conjunto que llamaremos *espacio de texto en claro* $\mathcal{M}$ y por último. el conjunto conformado por todos los textos cifrados lo nombraremos *espacio de texto cifrado* $\mathcal{C}$

Con fundamento en lo anterior, se observa que el algoritmo de cifrado utiliza alguna llave k en particular para definir el tipo de transformación que sufrirá *un* texto en claro. Concluimos que el número de transformaciones distintas para algún mensaje M está dado por la cantidad de elementos del conjunto $\mathcal{K}$.

Para identificar una función de cifrado específica utilizaremos el símbolo E_k y con ello se entenderá que k es alguna de las tantas llaves que se encuentra en $\mathcal{K}$. Puesto que esta función convierte un texto en claro M en uno cifrado C, la notación empleada para referirnos a este hecho es:

$$E_k : M \rightarrow C.$$

Una vez que se posee el texto cifrado C, para obtener el texto M se requiere de otro algoritmo que llamaremos *algoritmo de descifrado* D que, del mismo modo, a partir de una función definida por una k particular revertirá el proceso. Esta función, que se denotará como D_k. tiene como tarea realizar los cambios convenientes para obtener el texto en claro M, tomando como base el texto cifrado C y utilizando la llave k adecuada (Figura 1.3). Análogamente escribiremos:

$$D_k : C \rightarrow M$$

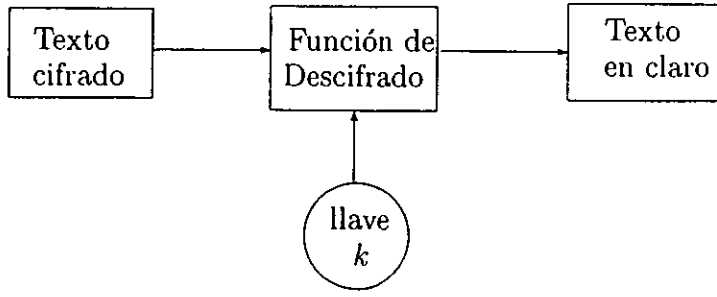


Figura 1.3: Descifrado de un mensaje

Por lo que hemos visto, el texto cifrado se confecciona mediante el texto en claro después de aplicarle una función de cifrado particular definida mediante cierta k escogida; es decir $C = E_k(M)$. De igual modo, en el caso inverso, M se obtiene de C usando una función de descifrado con la k adecuada; siguiendo la misma idea: $M = D_k(C)$

La razón de utilizar las letras E y D para identificar dichos algoritmos se basa en la traducción al Inglés: cifrado como *Encrypt* y descifrado como *Decrypt*. En la diversa literatura [STA95, ROB82, SIN66, PIN93, STI95], se utilizan las primeras letras del significado en Inglés y para ser consistentes, en nuestro estudio utilizaremos la misma notación.

Las diversas funciones criptográficas se encuentran clasificadas bajo diversos criterios, optaremos, por conveniencia, en emplear la clasificación basada en el *número de llaves*, que a su vez se subdivide en [STI95]:

- **Criptografía de Llave Secreta**: La llave utilizada para cifrar y descifrar es la misma.
- **Criptografía de Llave Pública**: Se utiliza una llave para cifrar y otra para descifrar.
- **Funciones Hash**: No requiere de alguna llave.

Es importante que ilustremos el papel que juega una llave porque además de ser el criterio de clasificación que hemos escogido también hemos observado que gracias a ésta se define una transformación única para un texto dado.

Haciendo una analogía, la llave juega el mismo papel que el de la combinación de una caja fuerte. Si uno conoce la combinación es posible abrirla, de igual forma, cuando uno conoce la llave es fácil obtener M o C, pero si se desconoce su valor debe de ser difícil obtener alguno de los textos, al igual que es complicado abrir la caja fuerte sin su combinación.

Hablando ahora de la otra rama de la criptología, el criptonálisis, cuya tarea consiste en conocer el mensaje M o la llave k, tiene diversas formas de operar, las cuales llamaremos *ataques*, que de igual forma se tienen clasificados [STI95] y en la sección 1.8 los retomaremos:

- **Sólo texto cifrado**: El criptoanalista obtiene varios textos cifrados y trata de inferir la llave k o su correspondiente texto en claro.
- **Texto en claro conocido**: El criptoanalista obtiene algun texto en claro con su correspondiente cifrado y busca inferir la llave utilizada para predecir un nuevo texto cifrado interceptado.
- **Texto en claro escogido**: A partir de un texto en claro se conoce su texto cifrado y con ello se busca determinar los efectos de la llave involucrada para determinar el texto en claro de un nuevo texto cifrado interceptado.
- **Función de cifrado conocido**: Mediante el funcionamiento de E_k se desea encontrar la función D_k.
- **Texto cifrado escogido**: Mediante textos cifrados y descifrados, donde el texto cifrado es el punto de inicio, la labor consiste en determinar la llave k.

Es importante el criptoanálisis, puesto que gracias a éste ha sido posible encontrar el texto en claro sin el conocimiento de la llave o deducir la llave a partir del texto cifrado, teniendo como resultado en cualquier caso el mensaje original.

Al trabajo del criptoanalista es común que se le identifique con el termino de *romper un criptosistema*. Para entender la razón de dicha frase haremos otra analogía: cuando un jarrón se rompe se obtienen fragmentos que ya no cumplen con su finalidad primordial: contener un líquido. Similarmente, un criptosistema cuando es analizado y es posible conocer M a partir de C sin el conocimiento de k, implica que el algoritmo de cifrado ya no está cumpliendo con su objetivo: ocultar la información.

Lo que significa para un artesano fabricar un jarrón más resistente, en la criptografía se refleja en el diseño de un algoritmo más robusto. Precisamente, gracias a la interacción que ha existido entre la criptografía y el criptoanálisis ha sido posible que los sistemas secretos evolucionen, dando como resultado que la criptología progrese.

Por último, es importante mencionar las ventajas obtenidas con el uso de la criptografía, puesto que a lo largo de nuestro trabajo estaremos haciendo referencia a ellas. A través del *modelo de referencia OSI*, que define una arquitectura de seguridad ("Information Processing Systems. OSI Reference Model - Part 2: Security Arquitecture", ISO/IEC IS 7498-2) [PIN93] se proponen los elementos necesarios para proteger las comunicaciones de los usuarios a través de una red, los cuales se les conoce como *servicios de seguridad*:

1. **Autenticación**: Verificar la fuente de los datos. La autenticación puede ser por parte del emisor, del receptor o de ambos.

2. **Control de acceso**: Verificar que los recursos son utilizados por quien tiene derecho a hacerlo.

3. **Disponibilidad**: Garantizar que a un usuario legítimo no se le niege el acceso al sistema.

4. **Confidencialidad** : Evitar que se revelen, deliberada o accidentalmente, los datos de una comunicación.

5. **Integridad**: Verificar que los datos de una comunicación no se alteren, esto es, que los datos recibidos por el receptor coincidan por los enviados por el emisor.

6. **No repudio**: Proporcionar la prueba, ante una tercera parte, de que cada una de las entidades en una comunicación han participado.

1.2 Criptosistema

Para definir un criptosistema necesitamos hablar del contexto en el que éste operará: para ello debemos mencionar que al comunicar algún mensaje se requerirá de un canal o medio de transmisión por el que viajará el mensaje; nosotros suponemos que éste es inseguro, lo que se traduce en la posibilidad de obtener o modificar al menos una parte del mensaje sin tener autorización para hacerlo.

Los canales inseguros se considerán así a raíz de que existe mucha gente que los utiliza, razón por la cual también se les conoce como *canales públicos*. Si omitimos que el mensaje viajará a través de un canal inseguro entonces no será necesario emplear técnicas para ocultar la información, puesto que no existe la posibilidad de extraer o alterar su contenido.

Hoy en día es indispensable hacer uso de las líneas telefónicas, redes de datos y un sinnúmero de canales donde es viable obtener o modificar parte de los mensajes que no son de nuestra propiedad, por lo que los canales inseguros son ineludibles en la mayoría de las comunicaciones actuales.

Considerando los parrafos anteriores y los los conceptos que introducimos en la sección 1.1, estamos preparados para presentar formalmente la definición de un criptosistema [ROB82]:

1. **Espacio de Texto en claro**, $\mathcal{M}$: Representa el conjunto de todos los textos en claro.

2. **Espacio de Texto cifrado**, $\mathcal{C}$: Representa el conjunto de todos los textos cifrados.

3. **Espacio de Llave**, $\mathcal{K}$: Representa el conjunto de todas las llaves factibles.

4. **Familia de funciones de cifrado**, $E_k : M \rightarrow C, k \in \mathcal{K}$: Representa las diversas funciones que son factibles para cifrar un texto claro.

5. **Familia de funciones de descifrado**, $D_k : C \rightarrow M, k \in \mathcal{K}$: Representa la gama de funciones viables para descifrar algún texto cifrado.

6. E_k y D_k se relacionan de tal forma que: $D_k(E_k(M)) = M$: Ilustra que la función de descifrado es la función inversa de la función de cifrado.

Para comprender la definición de criptosistema, analicemos la interrelación que hay entre cada uno de sus elementos, es decir, hay que puntualizar que del espacio de texto en claro, que se entiende como el conjunto de mensajes concebibles, se escogerá un mensaje en particular, el cual podrá ser modificado por una gama de funciones de cifrado E_k. La trasformación particular que sufrirá un mensaje dependerá de la llave k elegida dentro de todas las factibles agrupadas en $\mathcal{K}$.

Si nuestra tarea es aplicar todas las E_k a cada elemento de $\mathcal{M}$ obtenemos un conjunto conformado por todos los textos cifrados realizables, conjunto que llamamos espacio de texto cifrado $\mathcal{C}$.

Gracias a las funciones de cifrado un mensaje será oculto para aquellas personas no autorizadas, mientras que el destinatario auténtico si deseára obtener el texto en claro correspondiente, utilizará una función de descifrado, lo cual solamente es viable mediante una sola llave.

Cuando se intercambian mensajes cifrados, donde sólo los interesados auténticos conocen C, diremos que existe confidencialidad en dicha comunicación.

Hasta aquí hemos explicado cómo los puntos del 1 al 5 de la definición de criptosistema se relacionan. Para reforzar nuestros argumentos, en la Figura 1.4 ilustramos el esquema de cualquier criptosistema; observamos que el emisor, que lo denotamos con la letra A, envía un mensaje M por un canal inseguro y para ello es necesario aplicar ciertas alteraciones realizadas por la función de cifrado E_k. El receptor, quien es B, para conocer el mensaje M mediante C, necesitará de una función de descifrado D_k empleando la llave k correspondiente.

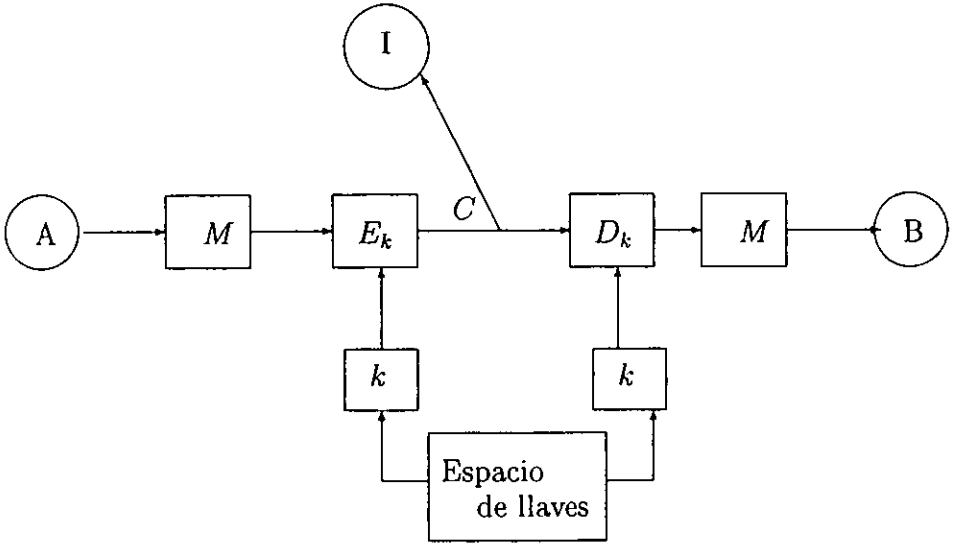


Figura 1.4: Criptosistema

Es importante señalar que las llaves involucradas no necesariamente son las mismas, en las secciones 1.5 y 1.6 abordaremos los casos particulares.

También hemos marcado con la letra I a lo que nombraremos *Intruso*, que en nuestro contexto equivale al criptoanalista, quien intenta obtener el mensaje cifrado, debido a que el canal es público.

Simplificando lo que hemos explicado, mediante una notación compacta: si alguien envía de manera segura un mensaje a otra persona entonces ésta necesitará cierta $k \in \mathcal{K}$ y determinará $C = E_k(M)$, $M \in \mathcal{M}$, $C \in \mathcal{C}$; y el destinatario, quien recibe C, para obtener el mensaje original calcula $M = D_k(C)$ con la $k \in \mathcal{K}$ correcta. Si el destinatario ignora k, no conocerá M y se encontrará en la situación de aquellos receptores no auténticos, que llamamos intrusos, cuyo objetivo es conocer M sin conocer la k correspondiente.

Con todo lo mencionado anteriormente, es necesario garantizar que ambas funciones deben de ser eficientes para todas las llaves escogidas, lo que significa: Una vez aplicada $E_k(M)$, si alguna persona conoce C, debe de serle complicado obtener M sin el conocimiento de k, mientras quien conoce k debe determinar M fácilmente. Esta proposición debe de ser válida para toda $k \in \mathcal{K}$.

Aunque durante un tiempo [KAH67] la gente consideró que era conveniente mantener en secreto la forma mediante la cual E_k y D_k funcionaban, hoy en día sabemos que no es necesario debido a la creciente necesidad de establecer comunicaciones seguras con un gran número de organizaciones.

La espina dorsal de un criptosistema es cierto planteamiento, principalmente matemático, que es complicado de resolver y ampliamente estudiado por toda la comunidad científica, lo que significa que el criptosistema es aceptado hasta que el problema es resuelto de manera práctica y eficiente.

Es importante considerar que de la infinidad de problemas, a un criptosistema le interesan aquellos cuya solución es complicada siempre y cuando no se conozca alguna pista. Es permisible aventurarnos a decir que la llave juega el papel de pista, es decir, la seguridad de un criptosistema se basa en mantener en secreto la llave y no el funcionamiento de E_k y D_k.

No hay que olvidar que si alguna persona logra resolver el problema sin el conocimiento de la pista, entonces será necesario auxiliarse de algún otro. Esta situación no es alarmante, puesto que la ciencia cuenta con un sinnúmero de problemas que son complicados de resolver aún sabiendo cuáles son.

Un ejemplo, que retomaremos en la sección 2.4, es a partir de la expresión $\beta = \alpha^k$ donde si únicamente conocemos α y β entonces la función logaritmo nos auxiliará a encontrar k, sin embargo mediante otras consideraciones éste problema es sumamente complicado de resolver en un tiempo corto. Como observamos, un problema tan inocente en un inicio se transforma en otro sumamente complejo, es aquí donde la creatividad es fundamental en el diseño de criptosistemas.

Para finalizar, si en este momento exigimos que además las funciones de cifrado y descifrado deben ser fáciles de usar, la intuición nos diría que es mucho pedir porque nos imaginamos que si un problema complicado es la base de una gama de funciones, entonces la forma de operar de ellas ha de ser difícil también. La realidad es que gracias a las matemáticas esta propuesta no es absurda, por lo que su uso no significará una tarea complicada y a su vez brindará seguridad a la información gracias a que el problema en el que se basa es complicado.

A medida que avancemos en nuestro trabajo introduciremos algunos de los problemas matemáticos de interés para la criptografía, siendo así como quedarán más claras estas afirmaciones.

Es importante señalar, que aunque en múltiples ocasiones mencionamos que existen problemas complicados de resolver, no hemos aclarado si la dificultad va orientada hacia una computadora, un grupo de personas o un niño de 6 años. La respuesta es que nos interesan los problemas que son complicados de resolver mediante algún sistema de cómputo.

Para aseverar que un problema es complicado o mejor dicho, computacionalmente complicado, es necesario garantizarlo mediante el estudio de la *complejidad computacional*, dicho campo de estudio de igual forma requiere de otras ideas matemáticas.

Las matemáticas de nuestra investigación son ya extensas como para aventurarnos a incorporarle las necesarias para justificar la complejidad computacional de cada una de las funciones de cifrado que analizaremos, es por ello que la omitimos. Sin embargo, si se desea profundizar en dicha línea de trabajo sugerimos consultar [ROS93, ROB82].

Para pasar a la siguiente sección, hay que observar que no hemos justificado las razones por las cuales siempre el texto en claro se puede obtener a partir del texto cifrado con la correspondiente llave. Para responder dicho cuestionamiento es necesario comprender la expresión $D_k(E_k(M)) = M$, que se encuentra en el último punto de la definición de criptosistema, para lo cual nos remontaremos al significado preciso de una función, y precisamente en la siguiente sección abordaremos dicho tema.

1.3 Definición de Función

Para auxiliarnos en la explicación de la presente sección, haremos uso de la experiencia previa que adquirimos con los números reales, bajo esta advertencia una función es contemplada como una regla que asigna un elemento $x \in \mathbb{R}$ con otro elemento $y \in \mathbb{R}$, por ejemplo de funciones tenemos la relación que asigna a todo número su cuadrado o la que asocia a cada número x el número $x + 2$.

Hay que señalar que no necesariamente la regla debe estar dada por expresiones algebraicas, ni tampoco por una regla que tenga sentido práctico. Es válido asociar 2 a π, 3 a e, *perro* a *libro* aunque de primera impresión no veamos una razón en hacerlo.

Un diagrama común para expresar una función que relaciona un numero x con el y, el 3 con e y 2 con π es mediante la Figura 1.5.

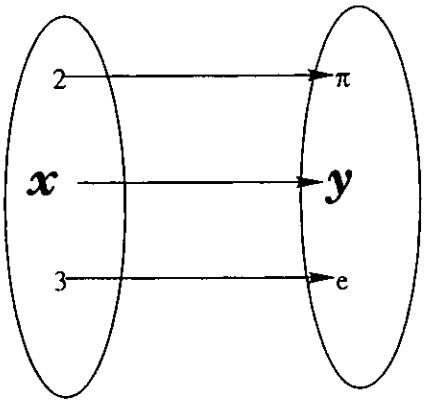


Figura 1.5: Función

El referirse a una "función" mediante sinónimos como "asociación" o "regla" no es correcto, si queremos estudiar el significado de una función no hay que utilizar palabras semejantes porque se llegan a situaciones confusas; por ejemplo, ¿cuáles son los efectos cuando nos saltamos la regla? o ¿la asociación a que clase de números es aplicable?. Como observamos dichos enunciados no tienen una respuesta debido a que el alcance de una función no es claro.

Por eso, una definición satisfactoria no debe conformarse de palabras afines. Como comenta Spivak [SPI92], en vez de querer definir un objeto es mejor buscar todos los recursos suficientes para comprenderlo, siguiendo su sugerencia nos acercaremos a una definición clara sin complicación alguna.

Consideremos el siguiente conjunto F, que reúne cierta información sobre la función $f(x) = x^2$

$$F = \{(1,1), (-1,1), (2,4), (-2,4)\}$$

Con este conjunto contamos con la información suficiente para conocer $f(1)$, ya que sólo hay que encontrar el par ordenado que inicie con 1 y después tomar el segundo número. De esta forma $f(1) = 1, f(-1) = 1, f(-2) = 4$. Ahora bien, si consideramos otro conjunto G

$$G = \{(1,7), (3,7), (2,5), (1,8), (8,4)\}$$

Es claro que $g(3) = 7$, $g(2) = 5$, $g(8) = 4$, pero no sabemos si $g(1) = 7$ o $g(1) = 8$. Visto desde otro punto de vista, si queremos definir una función a partir de pares ordenados debemos de imponer otras consideraciones, las cuales aparecen a la vista cuando observamos el conjunto G.

Definición de función [APO92]: Una función es un conjunto de pares ordenados en donde si (a, b) y (a, c) pertenecen al conjunto, entonces $b = c$.

Con la definición precisa de una función, a partir de pares ordenados como lo hemos hecho, observamos que no existe alguna ambigüedad como la presentada en el conjunto G.

Una costumbre para denotar el primer elemento del par ordenado es con la letra x y el otro elemento con la letra y, donde sabemos que $y = f(x)$, de tal forma que el conjunto de pares ordenados es de la forma $(x, f(x))$.

Existen un sinnúmero de expresiones para $f(x)$, sin embargo, debido a que todavía no contamos con las matemáticas suficientes, en las secciones posteriores analizaremos aquellas que son de principal interés para la criptografía.

Para continuar es importante introducir una definición más [SPI92]: el conjunto de números a los que se les aplica la expresión $f(x)$ recibe el nombre de *dominio*, mientras que el conjunto de números resultantes recibe el nombre de *imagen*.

Para denotar que una función f asigna un elemento $x \in \mathbb{A}$ del dominio otro $y \in \mathbb{B}$, utilizaremos la siguiente notación notación.

$$f : \mathbb{A} \rightarrow \mathbb{B}$$

El conjunto $\mathbb{B}$ lo llamaremos contradominio y es importante recalcar que tanto el conjunto $\mathbb{A}$ como el contradominio son conjuntos que *no* necesariamente son números.

En el mundo de las matemáticas el concepto de función es uno de los más importantes y usados, ya que partiendo de la definición de función es factible construir una gran variedad de funciones con caracteristicas muy particulares e interesantes. Para nuestro trabajo serán de interés enfocarnos en aquellas que se les conoce como funciones *inyectivas* y *suprayectivas*.

Las funciones *inyectivas* son aquellas que para dos elementos del dominio diferente les asocia un valor distinto en la imagen.

Para observar la naturaleza de las funciones inyectivas veamos un ejemplo de una función que no sea inyectiva:

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ f(x) &= x^2 \end{aligned}$$

La función mostrada **no** es inyectiva puesto que al tomar dos valores diferentes, $x_1 = 1$ y $x_2 = -1$, observamos que tienen el mismo valor en la imagen: $f(x_1) = f(x_2) = 1$.

Traduciendo las palabras que caracterizan una función inyectiva, mediante símbolos matemáticos, diremos que una función g es inyectiva, si para todo x_1 y x_2 elementos del dominio de g se tiene que:

$$g(x_1) = g(x_2) \Rightarrow x_1 = x_2$$

Por otra parte, las funciones *suprayectivas* consisten en aquellas que para todo elemento y del contradominio, existe un elemento x en el dominio de tal forma que $f(x) = y$. Un ejemplo para ilustrar la importancia de la definición es mediante la función f:

$$\begin{aligned} f &: \mathbb{Z} \to \mathbb{R} \\ f(x) &= x^2 \end{aligned}$$

El contradominio equivale a los números reales mientras que el dominio a los números naturales. Para afirmar que f es suprayectiva debemos asegurar que para todo elemento y en los números reales, existe un elemento x en los números enteros de tal forma que $f(x) = y$. Al analizar la función que hemos propuesto es claro que no es suprayectiva, para ello basta considerar a $y = \pi$, que es un número real, y observar que es imposible encontrar un entero x de tal forma que $x^2 = \pi$.

En las secciones siguientes nos interesarán aquellas funciones que cumplen con ambas propiedades, es decir, son inyectivas y suprayectivas las cuales las identificaremos con el nombre de *funciones biyectivas* [APO92].

Por último, vamos a considerar una de las operaciones entre funciones que son de nuestro interés: *composición*. Para comprender esta idea consideremos dos funciones, una g que relaciona el conjunto $\mathbb{A}$ con el $\mathbb{B}$ y otra función f que relaciona el conjunto $\mathbb{B}$ con el $\mathbb{C}$, es decir:

$$\begin{aligned} g &: \mathbb{A} \to \mathbb{B} \\ f &: \mathbb{B} \to \mathbb{C} \end{aligned}$$

Una pregunta que motiva la composición de funciones es: ¿será posible encontrar una función $h : \mathbb{A} \to \mathbb{C}$?. La respuesta es **sí** y precisamente se representa mediante $h = (f \circ g)(x) = f(g(x))$ con $x \in \mathbb{A}$ que se lee como "f compuesta con g ".

No nos debe extrañar dicha operación porque finalmente la función g evaluada en un punto x del dominio tiene un elemento $y \in \mathbb{B}$, mientras que los elementos del dominio de f son precisamente elementos del conjunto $\mathbb{B}$, por lo tanto $f(y)$ tiene sentido.

Apoyándonos con la composición de funciones supongamos la siguiente situación:

$$\begin{aligned} g &: \mathbb{A} \to \mathbb{B} \\ f &: \mathbb{B} \to \mathbb{A} \\ (f \circ g) &: \mathbb{A} \to \mathbb{A} \end{aligned}$$

Además vamos a definir la regla de correspondencia para $x \in \mathbb{A}$ y $y \in \mathbb{B}$ como:

$$g(x) = y \text{ cuando } f(y) = x$$

Para que la siguiente afirmación tenga sentido consideremos a la función g y f biyectivas [SPI92]:

$$\begin{aligned} (f \circ g)(x) &= f(g(x)) \\ &= f(y) \\ &= x \end{aligned}$$

A la función $h(x) = x$ la llamaremos *función identidad*, que para nuestro caso $h(x) = (f \circ g)(x)$, y cuando la composicón de función da por resultado a la función identidad concluimos que la función $g(x)$ es la *función inversa* de $f(x)$.

Para ilustrar el concepto anterior, consideremos una función f y g. ambas funciones biyectivas, definidas del siguiente modo:

$$\begin{aligned} f &: \mathbb{R} \to \mathbb{R} \\ f(x) &= \tfrac{x}{2} \end{aligned}$$

$$\begin{aligned} g &: \mathbb{R} \to \mathbb{R} \\ g(x) &= 2x \end{aligned}$$

$$\begin{aligned} f(g(x)) &: \mathbb{R} \to \mathbb{R} \\ f(g(x)) &= f(2x) \\ &= \tfrac{2x}{2} \\ &= x \end{aligned}$$

El material alrededor de las funciones es extenso, sin embargo para nuestros fines las definiciones que hemos introducido serán de gran utilidad más adelante. Para realizar una reflexión más profunda relacionado al tema que hemos tratado suguerimos consultar: [APO92, SPI92].

1.4 Enfoque funcional de un Criptosistema

Para nuestro propósito, definamos al conjunto de letras del alfabeto castellano como los elementos que producirán un mensaje cualquiera. por lo tanto, el espacio de texto en claro $\mathcal{M}$ será el conjunto compuesto por unidades de mensaje producidas por todas las factibles concatenaciones de dichos símbolos. No es necesario preguntarnos si todos los elementos del conjunto tienen un motivo para existir, por ejemplo, JKDL es un mensaje admisible aunque en nuestro lenguaje no tenga significado, pero si será importante considerar que cada mensaje de $\mathcal{M}$ es de longuitud finita.

Precisamente $\mathcal{M}$ será el dominio para nuestras funciones de cifrado. Aunque el contradominio para las funciones de descifrado es igual a $\mathcal{M}$, para no causar confunsión en la notación utilizaremos la letra $\mathcal{C}$. Es decir, el contradominio es el conjunto $\mathcal{C}$ que está formado por el conjunto de todos los mensajes posibles, a partir del conjunto de letras del alfabeto. de cierto tamaño.

Ya que estamos considerando el dominio y el contradominio como conjuntos finitos, con el mismo número de elementos, por ejemplo si nos preguntamos por todos los mensajes de longitud 3, y consideramos que nuestro alfabeto contiene 26 símbolos, entonces el espacio de texto en claro y cifrado tienen 26^3 mensajes diferentes.

Los símbolos utilizados en un mensaje pueden ser muy diversos, por ejemplo, si utilizamos el 0 y el 1, y además cada mensaje es de longitud 2, el espacio de texto en claro será $\mathcal{M} = \{00, 01, 10, 11\}$.

Como cada unidad de mensaje se encuentra formada por la concatenación de símbolos, entonces para un $M \in \mathcal{M}$ se definirá del siguiente modo:

$$M = \{m_1, m_2, ..., m_N\},$$

Donde el subíndice N hace referencia al tamaño del texto y cada m_i es algún símbolo válido utilizado en la construcción de cada mensaje. Por ejemplo, el mensaje "JDKL" define el siguiente conjunto:

$$M = \{J, D, K, L\}$$

Donde $m_1 = J$, $m_2 = D$, $m_3 = K$, $m_4 = L$, y dicho mensaje pertenece al espacio de mensajes en claro cuando $N = 4$ y hemos considerado los símbolos del alfabeto castellano.

Con la definición de función, que hemos introducido a partir de pares ordenados, en nuestro contexto una función de cifrado es aquella que impide que a dos textos en claro iguales se les asocie dos textos cifrados diferentes, es decir, si consideramos un $M \in \mathcal{M}$ la función de cifrado E_k define el siguiente conjunto EK de pares ordenados:

$$EK = \{(m_1, c_1), \ldots, (m_N, c_N)\}$$

donde:

$$\text{Si} \quad (m_i, c_i), (m_i, c_j) \in EK \Rightarrow c_i = c_j$$

La segunda entrada de cada par ordenado del conjunto EK determina el conjunto:

$$C = \{c_1, \ldots, c_N\},$$

Lo cual lo denotamos en un principio como texto cifrado. Para conocer cada uno de los elementos del conjunto anterior sólo se requiere calcular: $c_i = E_k(m_i)$.

Aunque no es crucial estudiar alguna función en particular, en la cripografía requerimos de ciertas restricciones adicionales, por ejempo si una función de cifrado E_k determina el conjunto $E1$ de pares ordenados:

$$E1 = \{(m_1, c_1), (m_2, c_1), (m_4, c_3), (m_3, c_5)\}$$

Aunque no se viola la definición de función es claro que dos unidades de texto, m_1 y m_2 diferentes, tienen el mismo texto cifrado c_1, lo que nos llevará a un error cuando pretendamos obtener M: no se sabrá si c_1 corresponde a m_1 o m_2.

Por lo tanto, las funciones de cifrado deberán impedir que sea concebible asignarle a un mismo texto cifrado dos textos en claro diferente, en una notación matemática escribimos esta afirmación como:

$$E_k(m_i) = E_k(m_j) \Rightarrow m_i = m_j$$

En la sección 1.3 observamos que al cumplir la propiedad anterior, afirmamos que dicha función lleva el nombre de inyectiva.

Por último, tambien es necesario considerar que, para todo $c_i \in C$ exista un $m_i \in M$ de tal forma que $E_k(m_i) = c_i$, dicho en otros términos, es necesario garantizar que todo $c_i \in C$ debe de ser el resultado de algún m_i después de aplicar nuestra función de cifrado E_k. Esto con el propósito de asegurar que la persona que obtenga C mediante la k adecuada determine M utilizando $D_k : C \to M$.

De igual forma, afirmamos en la sección 1.3, que al cumplirse la propiedad anterior, dicha función lleva el nombre de función suprayectiva.

El buscar que la función de cifrado sea inyectiva y suprayectiva es con el objetivo de garantizar que a partir del texto cifrado sea posible encontrar el texto en claro. Es decir, gracias a que la función de cifrado es biyectiva entonces garantizamos la existencia de una función inversa [CAR90], la cual justamente es D_k.

Mediante la función inversa de igual forma describimos un conjunto de pares ordenados definidos por el conjunto DK :

$$DK = \{(c_1, m_1), \ldots (c_N, m_N)\}$$

Donde cada elemento m_i está definido como $m_i = D_k(c_i)$. Y puesto que la función de cifrado es biyectiva entonces la función de descifrado también lo es [SPI92], y por lo tanto tampoco existirán errores en la asignación inversa.

Gracias a que las funciones de cifrado y descifrado son biyectivas, el siguiente paso es preguntarnos sobre el resultado cuando se realiza la composición de funciones entre la función de descifrado y de cifrado, es decir, determinar $D_k(E_k(M))$ y dado que:

$$\begin{aligned} M &= D_k(C) \\ C &= E_k(M). \end{aligned}$$

Con lo visto en la presente sección y en la 1.3, tenemos que:

$$\begin{aligned} E_k &: M \to C \\ D_k &: C \to M \\ (D_k \circ E_k)(M) &: M \to M \\ D_k(E_k(M)) &= D_k(C) \\ &= M \end{aligned}$$

Es decir, la función inversa para E_k es la función D_k y gracias a ésto garantizamos la posibilidad de encontrar el texto en claro a partir del cifrado.

1.5 Criptografía de Llave Secreta

La criptografía de llave secreta, conocida también como *criptografía de llave simétrica* o *criptografía convencional*, fué el primer método que la humanidad diseñó y utilizó para proteger los mensajes que intercambiaban [PIN93].

La criptografía de llave secreta se fundamenta en la necesidad de cifrar y descifrar un mensaje utilizando *una misma* llave, para comprender su funcionamiento supongamos que un emisor A busca comunicarse con un receptor B de una forma segura, para lo cual:

1. A y B acuerdan una llave $k_{ab} \in \mathcal{K}$
2. A y B mantienen k_{ab} en secreto
3. Si A o B desean transmitir un mensaje M, de una manera segura. entonces comunicarán $C = E_{k_{ab}}(M)$
4. Mediante el texto cifrado, M se determina como $M = D_{k_{ab}}(C)$

Como observamos, los interlocutores comparten una misma llave, denotada como k_{ab}, la cual es utilizada tanto para cifrar $(C = E_{k_{ab}}(M))$, como para descifrar $(M = D_{k_{ab}}(C))$.

Para ilustar los pasos que han seguido tanto A como B, nos remitiremos a la Figura 1.6 en donde es importante observar que *una* sola llave interviene en el proceso de cifrado y descifrado.

Ya que C es un texto cifrado, éste es indescifrable para aquellos que no cuenten con la llave k_{ab}, puesto que no es viable determinar $M = D_{k_{ab}}(C)$, con ésto garantizamos que un mensaje sea descifrado por aquellos individuos que compartan k_{ab}, lo cual se traduce en la confidencialidad de la comunicación entre A y B.

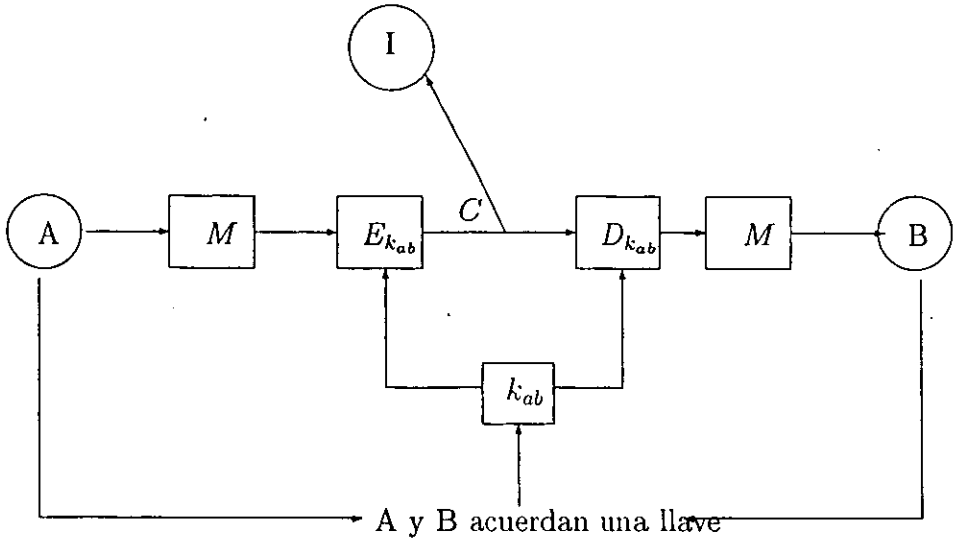


Figura 1.6: Esquema de la criptografía de llave secreta

Por otra parte, si por alguna circunstancia algún $c_j \in C$ es alterado, por decir en c'_j, cuando el receptor aplique la función $D_{k_{ab}}(C)$ se tendrá que:

$$\begin{aligned} M &= D_{k_{ab}}(C) \\ &\Rightarrow \text{ para algún } m_j = D_{k_{ab}}(c_j) \neq m'_j = D_{k_{ab}}(c'_j) \\ &\Rightarrow M \neq M' \end{aligned}$$

Como M' es diferente a M es probable que el significado sea distinto, por lo tanto el receptor asegurará que el mensaje recibido ha sido alterado y supondrá que las causas del daño se deberán a intervenciones de un intruso o porque durante el trayecto del mensaje se perdió o se alteró parte de éste.

Cuando un mensaje no sufre cambio alguno diremos que se tiene *integridad* en la comunicación. Para reflexionar en su importancia, veamos un caso muy simple: El acordar una cita para el 01/01/2000 es muy diferente al 01/01/2009, y con el hecho de haber alterado un digito las repercusiones pueden ser muy costosas.

Es importante indicar que cualquier señal pierde su intensidad durante su trayecto por el medio de comunicación, lo que origina que una fracción del contenido del mensaje se pierda o se altere, por ejemplo, un mensaje se percibe claramente cuando éste es comunicado al oído en vez de gritarlo a través de una gran distancia de separación.

Por último, si A y B quisieran verificar la identidad de ambos, es decir, B asegurarse que A es quien dice ser y A convencerse de que B es quien dice ser, entonces *una* posible solución es mediante el siguiente razonamiento:

- A remite M_1 a B
- B recibe M_1 y envía la pareja de mensajes :

$$(E_{k_{ab}}(M_1), M_2) = (C_1, M_2)$$

- A obtiene la pareja (C_1, M_2). Si $D_{k_{ab}}(C_1) = M_1$ entonces, A está seguro que B es quien dice ser.
- A manda $E_{k_{ab}}(M_2) = C_2$ a B
- B recibe C_2, si $D_{k_{ab}}(C_2) = M_2$ entonces. B está seguro que A es quien dice ser

Básicamente a través de la historia la confidencialidad fué lo más importantes en una comunicación [KAH67], en el caso de la autenticación de los interlocutores y la integridad de los mensajes éstos son resueltos fácilmente gracias a las matemáticas.

1.5.1 Ventajas y desventajas

La principal virtud de los criptosistemas de llave secreta se traduce en el poco tiempo que hay que invertir para cifrar y descifrar un mensaje cualquiera, por lo que resulta atractivo esta propiedad cuando hay que cifrar grandes volumenes de datos. Sin embargo, es mucho más interesante, para comprender la importancia de la criptografía de llave pública, analizar su principal defecto que es llamado: *intercambio de llaves.*

Para comprender el problema, consideremos un conjunto de n individuos definido del siguiente modo: $IND = \{I_1, I_2, ..., I_n\}$. Supongamos que cada elemento del conjunto anterior quiere comunicarse con el resto del conjunto, de donde se observa que:

- I_1 desea comunicarse con $I_2, I_3, \ldots, I_n$. I_1 requiere $\boldsymbol{n-1}$ llaves diferentes (I_1 administra $n-1$ llaves)
- Por el punto anterior, I_2 ya se ha comunicado con I_1 y le resta $I_3, I_4, \ldots, I_n$. I_2 ahora sólo requiere $\boldsymbol{n-2}$ llaves (I_2 administra $n-1$ llaves)
- En este momento a I_3 le resta comunicarse con $I_4, I_5, \ldots, I_n$. I_3 necesitará $\boldsymbol{n-3}$ llaves (I_3 adminitra $n-1$ llaves)
- Continuando el mismo razonamiento, nos detenemos cuando a I_{n-1} sólo le faltará $\boldsymbol{n-(n-1)=n-n+1=1}$ llave. (I_{n-1} administra $n-1$ llaves)
- Con esto, I_n ya no necesita alguna llave adicional para estar comunicado con el conjunto IND (I_n administra $n-1$ llaves)

Por ejemplo, en una comunidad de 8 personas, cada individuo administrará a lo más 7 llaves y el grupo deberá acordar un criptosistema en el que el espacio de llaves sea mayor a 28 elementos, para garantizar que las llaves involucradas no se repitan.

Hacer la generalización del caso anterior se logra al observar un grupo de n personas, donde cada individuo administrará a lo más $n-1$ llaves y el total de llaves demandadas estará dado por la suma S, definida a partir de las llaves utilizadas por el conjunto IND, de donde observamos que:

$$\begin{aligned} S &= (n-1)+(n-2)+(n-3)+\ldots+1 \\ &= 1+2+3+\ldots+(n-1) \\ S+S &= \overbrace{n+n+n+\ldots+n}^{(n-1)veces} \\ 2S &= (n-1)n \\ &\Rightarrow S=\tfrac{n(n-1)}{2} \end{aligned}$$

Por ejemplo, un grupo con $n = 1,000$ individuos, cada uno deberá administrar a lo mas 999 llaves y el número de llaves mínimas está dado mediante la expresión S, es decir:

$$\begin{aligned} S &= \tfrac{n(n-1)}{2} \\ &= \tfrac{1,000(999)}{2} \\ &= 449,500 \text{ llaves distintas} \end{aligned}$$

El administrar 999 llaves es una tarea laboriosa, aunque en la actualidad los sistemas de cómputo facilitan esta tarea, después de estudiar la criptografía de llave pública veremos cómo este inconveniente es minimizado.

Aunque la principal problemática no es administrar a lo más $n-1$ y generar $\frac{n(n-1)}{2}$ llaves, vemos que la tarea de *acordar* $n-1$ llaves **sí** es un inconveniente, ya que si durante el intercambio de una llave, ésta es conocida por un intruso, entonces esta persona ajena podrá descifrar y cifrar cualquier mensaje en donde intervenga dicha llave.

De lo anterior concluimos que es importante no poner en riesgo la llave y el afianzar a lo más $n-1$ intercambio de llaves es una gran limitante, la cual queda eliminada en la criptografía de llave pública que estudiaremos en la sección 1.6.

1.5.2 Ejemplo de Criptosistema de Llave Secreta

Para concluir la presente sección, incluiremos un ejemplo de un criptosistema de llave secreta, que ilustrará los elementos que hemos desarrollado hasta el momento.

Por ahora no queremos mostrar la expresión para definir E_k y D_k, sin embargo supongamos que bajo $k = 3$ la función de cifrado tiene por objetivo realizar un corrimiento del alfabeto tres unidades hacia la izquierda, es decir, la letra "A" será ahora la letra "D" y así sucesivamente. Por supuesto la función de descifrado realiza el proceso inverso.

Utilizando la notación introducida en la sección 1.3, tenemos que $E_3(\mathrm{A}) = \mathrm{D}$, $E_3(\mathrm{B}) = \mathrm{E}$, $\ldots$, $E_3(\mathrm{Z}) = \mathrm{C}$ y si consideramos un mensaje con más de un caracter entonces la función de cifrado o descifrado se aplicará a cada uno de las letras involucradas, por ejemplo:

$$\begin{aligned} E_3(\mathrm{ABCD}) &= \{E_3(\mathrm{A}), E_3(\mathrm{B}), E_3(\mathrm{C}), E_3(\mathrm{D})\} \\ &= \{\ \mathrm{D}, \mathrm{E}, \mathrm{F}, \mathrm{G}\ \} \\ D_3(\mathrm{EFGH}) &= \{D_3(\mathrm{E}), D_3(\mathrm{F}), D_3(\mathrm{G}), D_3(\mathrm{H})\} \\ &= \{\ \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}\ \} \end{aligned}$$

Empleando la definición de función, estudiada en la sección 1.3, mediante pares ordenados el resultado de la función de cifrado y descifrado lo representaremos mediante el conjunto $E3$ y $D3$:

$$\begin{aligned} E3 &= \{(\mathrm{A,D}), (\mathrm{B,E}), (\mathrm{C,F}), (\mathrm{D,G}), (\mathrm{E,H}), (\mathrm{F,I}), \\ &\quad (\mathrm{G,J}), (\mathrm{H,K}), (\mathrm{I,L}), (\mathrm{J,M}), (\mathrm{K,N}), (\mathrm{L,O}), \\ &\quad (\mathrm{M,P}), (\mathrm{N,Q}), (\mathrm{O,R}), (\mathrm{P,S}), (\mathrm{Q,T}), (\mathrm{R,U}), \\ &\quad (\mathrm{S,V}), (\mathrm{T,W}), (\mathrm{U,X}), (\mathrm{V,Y}), (\mathrm{W,Z}), (\mathrm{X,A}), \\ &\quad (\mathrm{Y,B}), (\mathrm{Z,C})\ \} \\ D3 &= \{(\mathrm{D,A}), (\mathrm{E,B}), \ldots, (\mathrm{B,Y}), (\mathrm{C,Z})\ \} \end{aligned}$$

Es decir $E_3(\mathrm{A}) = \mathrm{D}$, $E_3(\mathrm{B}) = \mathrm{E}$, $\ldots$, $E_3(\mathrm{Z}) = \mathrm{C}$. y $D_3(\mathrm{D}) = \mathrm{A}$, $D_3(\mathrm{E}) = \mathrm{B}$, $\ldots$, $D_3(\mathrm{C}) = \mathrm{Z}$. El dominio y el contradominio para ambas funciones está dado por el conjunto:

$$\{\mathrm{A,B,C,D,E,F}\ \ldots,\ \mathrm{U,V,W,X,Y,Z}\ \}$$

Lo que implica que $E_3(\beta)$ no tiene sentido, $D_3(8)$ tampoco ni $E_3(a)$, porque $8, \beta, a$ no se encuentran definidos en el algún par ordenado.

Con lo anterior, al transmitir el mensaje $M = \{$ UNAM $\}$, mediante los datos anteriores, el texto cifrado C se obtiene del siguiente modo:

$$\begin{aligned} \mathrm{C} &= \{E_3(\mathrm{UNAM})\} \\ &= \{E_3(\mathrm{U}), E_3(\mathrm{N}), E_3(\mathrm{A}), E_3(\mathrm{M})\} \\ &= \{\ \mathrm{XQDP}\ \} \end{aligned}$$

El receptor, quien ha recibido C, mediante el conjunto $D3$ determina el mensaje original.

$$\begin{aligned} M &= \{D_3(\ \mathrm{XQDP}\ )\} \\ &= \{D_3(\mathrm{X}), D_3(\mathrm{Q}), D_3(\mathrm{D}), D_3(\mathrm{P})\} \\ &= \{\mathrm{UNAM}\} \end{aligned}$$

Con lo abarcado hasta el momento es tedioso determinar:

$$C = E_3(\ \mathrm{SIN\ MATEMATICAS\ SE\ SUFRE\ MAS}\ )$$

Sin embargo como veremos en la sección 2.2.2 seremos capaces de cifrar textos de longitudes mayores sin problemas.

1.6 Criptografía de Llave Pública

La criptografía de llave pública fue inventada por Whitfield Diffie y Martin Hellman en 1976 como una solución para el problema del intercambio de llaves presente en la criptografía de llave secreta (sección 1.5). La idea consiste en escoger en primer lugar una llave, llamada *llave privada*, que mediante algunas operaciones matemáticas se transformará en otra llave, denominada *llave pública*, dicho proceso se le identificará con el nombre de *generación* de llaves.

El éxito de la criptografía de llave pública radica en que es computacionalmente imposible extraer la llave privada a partir de la llave pública. Acordar alguna información secreta entre el receptor y el emisor queda eliminado: las comunicaciones seguras se efectuan con la llave pública y ninguna llave privada es comunicada o compartida, es decir, si deseamos comunicar un mensaje cifrado, entonces requerimos la llave pública del receptor, mientras que éste, puesto que cuenta con la llave privada correspondiente, será la única persona capaz de descifrarlo.

Como la llave pública y privada empiezan con p, haremos un cambio de notación. Considerando que la llave pública es empleada para cifrar, para el caso de B, la notación usada será e_b (*llave para cifrar los mensajes dirigidos a* B), mientras que la llave privada, imprescindible para descifrar un mensaje, para B escribiremos d_b (*llave de* B *para descifrar sus mensajes*).

La llave pública se conoce y no se mantiene en secreto; cuando alguien desea comunicar un mensaje, por decir a B, será necesario valerse de la función de cifrado E_{e_b}, involucrando la llave pública de B. así como vemos en la Figura 1.7.

El receptor auténtico, es decir B, para conocer el mensaje original utiliza la función de descifrado descrita a partir de su llave privada, es decir, D_{d_b}. Por lo anterior, el hecho de que cada individuo administre su propia llave privada para descifrar sus mensajes implica que únicamente hay que mantener en secreto *una* sola llave.

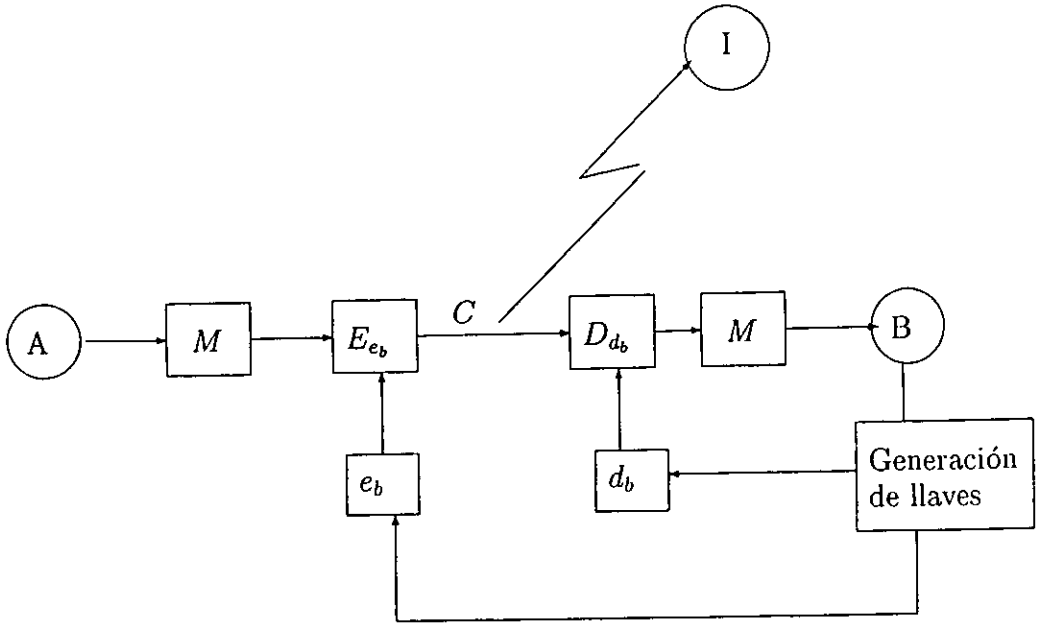


Figura 1.7: Criptografía de llave pública

Igualmente importante, debido a que las comunicaciones seguras hacen uso de la llave pública del receptor, quiere decir entonces que no hay que acordar alguna llave previamente, como en el caso de la criptografía de llave secreta (ver sección 1.5).

Por último, es importante señalar que los criptosistemas basados en la criptografía de llave pública desgraciadamente requieren de un mayor tiempo de cómputo para cifrar o descifrar algun mensaje. Aunque ésta es su principal inconveniente, introduce una virtud llamada *firma digital*.

1.6.1 Firma digital

Para motivar la justificación de las firmas digitales, consideremos el escenario de intercambio de mensajes cifrados entre Alicia y Beto, mediante un criptosistema de llave secreta.

Para entablar una comunicación segura, Alicia y Beto acuerdan una llave (k_{ab}) con la cual se definirá la función de cifrado y descifrado particular. Una vez hecho ésto, si en alguno de los mensajes Alicia se compromete jurídicamente con Beto, por decir, en la compra de un lote de 100 pares de zapatos mediante el contrato cifrado $C = E_{k_{ab}}(M)$ entonces es importante reflexionar sobre algunas consecuencias.

Si ambos son honrados no hay nada que añadir, sin embargo la mente humana en ocasiones es muy maliciosa y por ello conjeturemos que por motivos desconocidos Alicia reflexiona sobre dicho contrato y opta por no cumplirlo. Beto, que ha confiado en ella, ante esta situación interpone una demanda para que un juez la obligue a cumplir lo acordado.

El abogado de Alicia, que sabe criptografía, argumenta al juez: "Como ambos comparten una misma llave, entonces Beto ha fabricado un contrato en el que compromete a Alicia sin su autorización". Ante este razonamiento el juez no tendrá los argumentos adecuados para culpar a Alicia.

También es factible la situación inversa, ahora Alicia se apega al contrato pero es Beto quien se niega a cumplirlo. Esto lleva a Alicia a interponer una demanda en contra de Beto, ya que análogamente quiere obligar a Beto a responder por su parte.

El abogado de Beto observa que, dado que ambos comparten la misma llave, Alicia fué quien elaboró dicho contrato sin el permiso de Beto. Por lo tanto, no hay pruebas que culpen a Beto en su falta.

En los dos casos que hemos mostrado, no existen los elementos para emitir un veredicto justo, debido a que ambos comparten un mismo secreto.

Al firmar un contrado o convenio en donde exista un compromiso, ninguna de las partes deberá negarlo, en los documentos impresos existen abogados, jueces o notarios quienes tienen la facultad de proteger y custodiar el cumplimiento de los acuerdos legales por si éstos no son cumplidos; es así como los intereses de las partes quedan protegidos, y esta idea nos motiva en buscar *algo* que garantice lo mismo en los medios electrónicos.

Lo que buscamos es extrapolar las obligaciones jurídicas en un medio electrónico. Para resolver dicho problema, bajo la premisa de que la llave privada se encuentra en el poder de una sola persona, el dueño podrá *firmar* un documento haciendo uso de su llave y será válida la firma dado que es imposible inferir la llave privada a partir de la llave pública.

Por ejemplo, supongamos que el contrato M comunica una compra de un lote de 100 pares de zapato. Si Alicia quisiera comprometerse jurídicamente con Beto enviará $C = E_{d_a}(M)$. Como solamente Alicia cuenta con la llave d_a, ya no habrá duda de que ha sido ella quien ha mandado dicho mensaje y diremos entonces que Alicia a *firmado* M.

De forma genérica, como ilustramos en la Figura 1.8, cuando un emisor cifrar algún mensaje con su llave privada, decimos que C está firmado. Quien recibe el mensaje, como lo ilustramos en la Figura 1.9, empleará la llave pública del emisor para obtener el mensaje original y así verificará si pertenece al verdadero emisor.

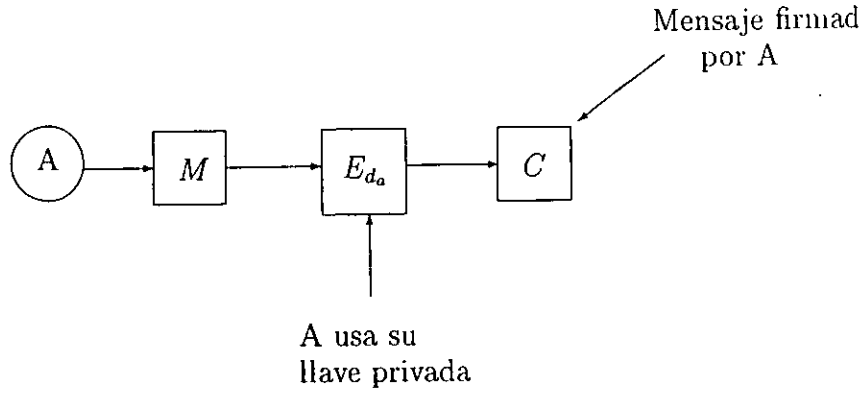


Figura 1.8: Firma de un mensaje por A

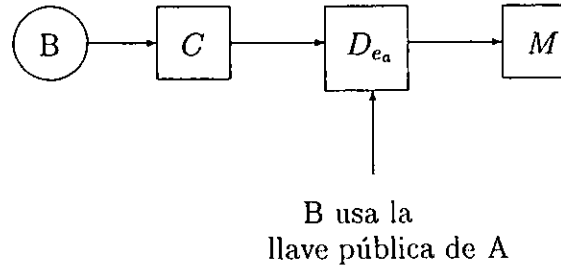


Figura 1.9: Verificación de la firma de A

1.6.2 Ventajas adicionales

De la Figura 1.9 concluimos que cualquier persona podrá conocer el mensaje que ha sido firmado, puesto que interviene la llave pública, la cual es conocida; ésto nos lleva a la necesidad de proteger la confidencialidad del mensaje.

Como la llave pública es esencial para cifrar los mensajes, para que A envie un mesaje M confidencial y firmado a B requerirá conocer la llave pública de B y con ello, en una primera etapa firmar el mensaje M:

$$C_1 = E_{d_a}(M)$$

Y para proteger la confidencialidad, entonces al mensaje firmado C_1, ahora lo cifra con la llave pública de B, es decir:

$$\begin{aligned} C &= E_{e_b}(C_1) \\ &= E_{e_b}(E_{d_a}(M)). \end{aligned}$$

B, quien recibe C, para conocer el mensaje original, descifrará el mensaje, en una primera tarea, mediante su llave privada y después con la llave pública del emisor que ha firmado el mensaje, es decir:

$$M = D_{e_a}(D_{d_b}(C))$$

Una forma gráfica de representar lo que hemos escrito con símbolos es con la Figura 1.10.

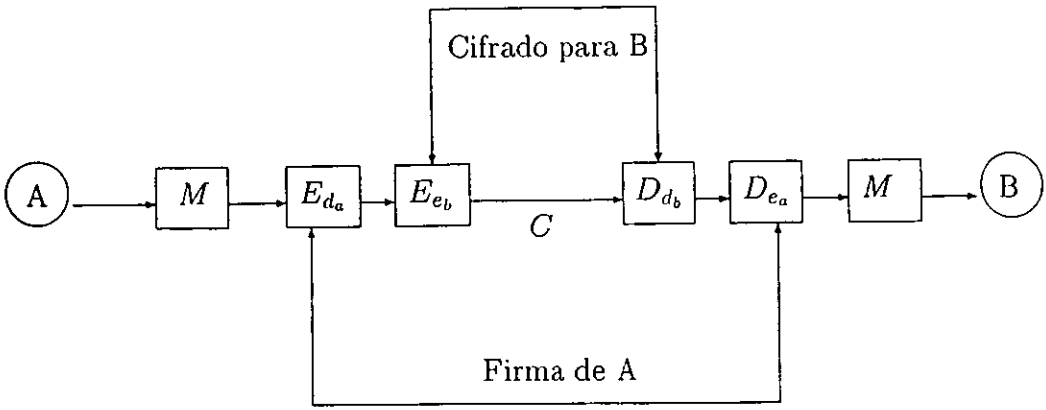


Figura 1.10: Mensaje firmado por A y cifrado para B

Para reflexionar sobre la importancia de la criptografía de llave pública, hagamos la siguiente observación: Una firma autógrafa es conocida por muchas personas, y además es factible falsificarla debido a puede existir una persona con la habilidad de realizar los mismos trazos que dieron origen a una rúbrica y con ello hacerse pasar por el auténtico dueño, ya que nadie firma un documento de diferente forma.

A diferencia de todo ésto, ya vimos que la *firma digital* consiste en cifrar los mensajes con la llave privada, por lo que para falsificarla sería necesario conocer la llave privada, sin embargo como ésta se encuentra en manos del dueño y es imposible deducirla a partir de la llave pública, obtener la llave privada de alguien es practicamente imposible. Adicionalmente, también es admisible que el mensaje firmado sea confidencial.

Para finalizar, queremos puntualizar que gracias a la criptografía de llave pública un mensaje además de ser auténtico también podrá ser confidencial, situación inalcanzable en un esquema no electrónico sin matemáticas, como sucede en los documentos impresos en papel.

Para mostrar el funcionamiento de algún criptosistema de llave pública en particular, con los fundamentos que estudiaremos en la sección 2.2 y 2.3 daremos un ejemplo que dejará claro lo mostrado.

1.7 Funciones Hash

En base a la clasificación de criptosistemas, indicada en la Sección 1.2, sólo nos resta hablar de las *funciones Hash* o *funciones de un sólo sentido.*

No es nuestro principal interés trabajar con estas funciones, puesto que no ayuda a cubrir los objetivos definidos, sin embargo es interesante preguntarse sobre la seguridad obtenida a través de su uso, sabiendo que no existe alguna llave involucrada.

En el análisis de funciones de la sección 1.3, mostramos los elementos a considerar para determinar la función inversa de una función f, es decir, si f es biyectiva entonces es posible construir "un sentido" inverso mediante otra función f^{-1}, de tal suerte que $f(f^{-1}(x)) = x$.

Al pensar en una función f de un sólo sentido, como es el caso de las funciones Hash, es equivalente a diseñar alguna función en la que es imposible determinar el sentido inverso f^{-1}.

Justamente en las funciones Hash es imposible crear un sentido inverso, es decir, al aplicar la función Hash (H) a un mensaje cualquiera, afirmaremos que es irrealizable encontrar cierta H^{-1} de tal suerte que $H(H^{-1}(M)) = M$.

Garantizar que no es viable encontrar el "sentido" que invierta los cambios realizados por $H(M)$ es similar a concluir que H no es inyectiva o suprayectiva.

La entrada de una función Hash consiste en un mensaje M, mientras que su salida es un valor h de longitud fija, es decir, $h = H(M)$. La construción de una función Hash es una tarea complicada, ya que su diseño debe de ser cuidadoso para evitar que de dos mensajes diferentes se obtenga un mismo valor h, es decir encontrar un M_1 y M_2 tales que $H(M_1) = h = H(M_2)$.

Gracias a que los valores h son distintos para cada mensaje significa que si por alguna razón el contenido del mensaje es modificado entonces el valor h ya no es semejante. De aquí deducimos que el objetivo de una función Hash es proteger la integridad del mensaje que es transmitido.

Al igual que las funciones de cifrado tuvieron que cumplir con ciertas restricciones, las funciones Hash también, para explicarlas basta considerar que el contradominio de la función es mucho menor que el dominio, lo cual significa que es posible encontrar dos mensajes M_1 y M_2 que tengan la misma función Hash, es decir, $H(M_1) = H(M_2)$, sin embargo, cuando una función Hash está bien definida, encontrar M_1 y M_2 es practicamente imposible.

Una forma de utilizar una función Hash en una comunicación segura, es mediante su cálculo antes de cifrar un mensaje M, es decir, obtener el valor h:

$$h = H(M)$$

Y posteriormente, comunicar la pareja h y M cifrados, es decir, transmitir la pareja:

$$(E_k(M), E_k(h))$$

Con ello, el receptor, una vez conocido el valor de $M = D_k(C)$ al determinar el valor Hash, deberá coincidir con el valor $D_k(h)$ recibido. Si el valor Hash no es idéntico entonces se tiene la certeza de que el mensaje ha sufrido alteraciones.

No discutiremos las matemáticas utilizadas para el diseño de alguna función hash, ni incluiremos algún ejemplo, ya que dichos elementos no son relevantes para los propósitos de nuestro trabajo, sin embargo para profundizar en el tema sugerimos: [ROB82, ROS93]

1.8 Criptoanálisis

El criptoanálisis es una tarea muy compleja a raíz de que no existe un método único para operar; prácticamente el éxito de un criptoanálisis depende de la habilidad del criptoanalista para encontrar los elementos suficientes y necesarios para descifrar un texto sin el conocimiento de la llave.

En un principio la creatividad del criptoanalista era la única herramienta, sin embargo ahora se requieren conocimientos en computación y matemáticas para lograr un criptoanálisis exitoso. La razón de ésto ha sido porque los criptosistemas hoy en día se basan en fundamentos matemáticos y de cómputo.

Por ejemplo, en el Siglo XVIII los métodos para cifrar un mensaje eran variados y aquellos criptoanalistas debían contar con una gran intuición para determinar exactamente la función de cifrado utilizada, al igual que de una gran paciencia para descifrarlo, puesto que no había un mecanismo mecánico que les auxiliara en su trabajo.

La importancia que toma el criptonálisis inició con el valor asociado a un mensaje, es por ello que en un inicio se relacionó a la criptología con ambientes militares y diplomáticos, en donde la información utilizada es de gran valor.

En la actualidad el valor de la información sigue existiendo, sin embargo, la diferencia con el pasado es que ahora se tiene un conocimiento amplio sobre el funcionamiento de los criptosistemas y se cuentan con sistemas de cómputo que pueden facilitar el trabajo.

Aunque en la actualidad existen las condiciones para efectuar un criptoanálisis existoso, uno de los problemas al que nos enfrentamos hoy en día se relaciona con el *costo* inherente al criptoanálisis, ya que aunque algunos criptosistemas pueden romperse, éstos requieren de inversiones millonarias o de una gran cantidad de tiempo.

De lo anterior concluimos que hoy en día las matemáticas y la computación juegan un papel relevante en el criptoanálisis y que además mientras que no se demuestre la inseguridad de un criptosistema, éste se seguirá utilizando.

Otro de los factores determinantes en el criptoanálisis se relaciona con el tipo de criptosistema, en el caso de la criptografía de llave pública el esfuerzo se concentra en la tarea de encontrar la llave privada a partir de la llave pública. Como observamos el criptoanalista no requerirá cierto texto cifrado.

En el caso de la criptografía de llave secreta, el criptoanálisis está vinculado con las propiedades de los criptosistemas. En nuestro trabajo haremos un criptoanálisis basado en un análisis estadístico, nuestra tarea será la de estudiar diversos textos cifrados conocidos, para obtener una conclusión que nos de la posiblidad de descifrar un texto cifrado desconocido.

A pesar de la gran diversidad inherente en el criptoanálisis, es importante mencionar algunos de los caminos por los que comúnmente se iniciará dicha labor, los cuales llamaremos ataques y representan un buen punto de partida en el criptoanálisis.

La situación más común a la que se enfrenta un criptoanalista es cuando solamente llega a sus manos el texto cifrado, ya sea porque fué posible extraerlo de algún dispositivo o fue escuchado cuando se comunicaba. Este tipo de ataques se llama *Sólo texto cifrado* [STA95].

Del ataque anterior, una posible solución inmediata es la de probar todas o algunas de las llaves hasta que encuentre un texto que ya no esté codificado. Por supuesto que si el criptoanalista en algún intento obtiene el mensaje original M que se encuentra en Ruso y el no sabe dicho idioma, el problema de entenderlo en sí será otra tarea para descifrarlo. Es por ello que se requiere tener los datos suficientes para el reconocimiento del texto en claro cuando éste se presente.

Claro es que la alternativa que hemos ilustrado no es la única, por ejemplo, el criptoanalista puede buscar cierta información que le ayude a limitar el número de intentos, es decir, al conocer la forma en la que se escoge k, el numero de intentos puede disminuir.

Por ejemplo si la llave está formada por 3 caracteres del alfabeto, el número de posibles llaves es de 26^3, sin embargo si el criptoanalista sabe que el primer caracter de la llave es la letra "B", entonces solo tendra que probar 26^2 llaves.

De algo tan simple, como es el ataque *solo texto cifrado*, hemos incluido dos caminos que se pueden seguir. Pero las alternativas son muy numerosas y es practicamente imposible discutir todas.

Una consideración importante para el ataque *solo texto cifrado* es la de contar con el texto cifrado suficiente, por ejemplo, para la palabra "ABCR" cifrada, existirán una variedad de textos en claro posibles, cuando no se cuente con la llave, como: "OLAS", "LUNA" o "AMOR", entre una infinidad de posibilidades.

Para demostrar dicho razonamiento la Teoría de la Información, que estudiaremos en la sección 2.1, nos dará los instrumentos necesarios para hacerlo, de igual forma gracias a su estudio sabremos cuánto texto cifrado (teóricamente) se requerirá para descifrar un mensaje sin el conocimiento de la llave.

Otro ataque importante es cuando el criptoanalista ingeniosamente obtiene la pareja de mensajes *(texto en claro,texto cifrado)*, por ejemplo, si se sabe de antemano, por lo menos, alguna palabra con la que empieza el texto en claro o porque días después, cuando la información ya no tuvo importancia, el mensaje cifrado es descifrado y conocido por el criptoanalista, entonces se conoce la forma en la que se realiza la asociación entre el texto en claro con el cifrado, dicho ataque lleva el nombre de *texto claro conocido*.

Un ejemplo concreto del ataque anterior, es observando al mecanismo de autenticación mostrado en la sección 1.5, en donde los participantes de una comunicación intercambiaban la pareja $(M, E_k(M))$, lo que daría la pauta para el ataque de *texto claro conocido.*

Por último, el mejor de los escenarios es cuando el criptoanalista a partir de un texto particular obtiene el texto cifrado. Por ejemplo supongamos a una compañía de telégrafos que ofrece un servicio de mensajes cifrados, y el proceso de cifrado y descifrado es como el estudiado en la sección 1.5.2.

Con el escenario anterior, si nos enviáramos un telegrama con las 26 letras del alfabeto obtendríamos la correspondencia entre el texto en claro y el texto cifrado, por lo tanto, si se interceptara cierto mensaje tendríamos los datos suficientes para descifrarlo. Este tipo de ataque se llama *Texto en claro escogido* [STI95].

Capítulo 2

Fundamentos Matemáticos

2.1 Teoría de la Información

El principal objetivo de esta sección es la de medir la cantidad de información inherente en una serie de mensajes; para lograrlo nos basaremos en nuestra interpretación intuitiva de "información", palabra que usamos a diario sin conocer claramente su significado.

Posteriormente dedicaremos nuestra atención en comprender un resultado que se desprende del análisis que produciremos, cuyo interés es esencial para el criptoanálisis: *Distancia de Unicidad.*

Para comenzar, imaginemos un artefacto tan simple como lo es el timbre de una casa, cuya finalidad consiste en transmitir dos mensajes: cuando es activado avisa la llegada de una visita y en caso contrario indica la ausencia de ella.

Si el timbre jamás es activado, entonces el único mensaje (ausencia de visitas) ya sería conocido por los dueños de la casa, y como no habría otro mensaje desconocido el timbre no tendría razón de existir.

Del ejemplo anterior, podemos notar que es factible relacionar el significado de "conocimiento" con el de "información" [MAS87]. Es decir, es similar "conocer" que hay una visita esperando en la puerta, con el de recibir la "información" sobre la llegada de un visitante a nuestra puerta.

Por lo tanto, debemos concentrarnos en la fuente que emite una serie de mensajes, que visto como un todo, conforma lo que llamaremos información. Para nuestro trabajo no nos enfocaremos en una fuente de información particular, por lo que seguiremos un análisis general.

El reto principal al que nos enfrentaremos será el de examinar lo que la fuente de información emite, es decir, si solamente transmite mensajes, entonces la meta será observar las propiedades que tienen estos mensajes para obtener una visión más clara de lo que la información es.

Para facilitar la comprensión de una definición de información, obervemos nuestra experiencia al comunicarnos con la gente que nos rodea; infinidad de veces nos hemos encontrado con la situación donde una amistad nos comunica cierta novedad que tiene un efecto en nosotros; por ejemplo, si además de ser nuestro amigo es un gurú en las finanzas y nos orienta sobre las acciones que incrementarán su valor, nos sorprenderemos un poco y seguramente tendremos la idea de que hemos adquirido cierta ventaja sobre el resto de los accionistas.

Cuando una plática similar a la que hemos mostrado tiene lugar en la vida real, es factible que la persona que reciba el mensaje crea que ha obtenido cierta *información*. La razón de ésto, es porque en la vida cotidiana asociamos a la información con el conocimiento o la novedad. Aunque esta definición de información es una muy buena aproximación [SIN82], nuestra labor ahora es definir una medida que justifique nuestro razonamiento informal.

El definir a la información basándose en ideas, que son igualmente complicadas de entender, como *novedad* o *conocimiento*, entorpece la tarea para proporcionar una métrica o medida. Es decir, a raíz de que que un mensaje o no es novedoso o no aporta cierto conocimiento para todas las personas que lo reciben, entonces es imposible definir en términos numéricos elementos que son propios de la interpretación de cada individuo.

A pesar de ello, nosotros seremos capaces de eliminar los elementos subjetivos y obtendremos como resultado un modelo matemático que cuantifique lo que entendemos por información.

Nuestra finalidad, para esta sección y la siguiente, es la de encontrar una medida que indique la cantidad de texto cifrado que se requerirá para encontrar el texto en claro correspondiente, sin el conocimiento de la llave, planteamiento que no justificamos cuando hablamos del ataque texto en claro conocido en la sección 1.8.

Para medir la cantidad de información partiremos de la premisa de que la información como tal existe, porque alguien la genera, al igual de que estudiaremos una fuente de información general, cuyo objetivo es la de emitir una serie de mensajes para comunicar cierta "novedad".

Iniciemos por hablar del caso más simple: una fuente de información que transmite dos mensajes, digamos el 0 y 1. Es importante señalar que no tendrá sentido preguntarse sobre la información de *un* mensaje aislado; cuando hablamos del timbre al inicio de esta sección, indicamos que si éste nunca se activara (o desactivara) no se comunicaría información adicional a la conocida.

El que escogamos al 0 y al 1 como dos posibles mensajes no limita a nuestra trabajo, prácticamete es permisible escoger cualquier otro símbolo, de igual forma, tampoco es importante considerar que un mensaje estará conformado por un símbolo exclusivamente, por ejemplo, si la fuente de información tiene la capacidad de transmitir mensajes con tres símbolos, entonces los mensajes posibles a comunicar serían:

$$\{000\}, \{001\}, \{010\}, \{011\}, \{100\}, \{101\}, \{110\}, \{111\}$$

No es de nuestro interés preguntarnos sobre la interpretación de cada uno de los mensajes anteriores y por lo tanto, temporalmente vamos a definir la cantidad de información como el número de mensajes posibles de cierta longitud que transmite una fuente de información. Del ejemplo anterior, la cantidad de información sería de 8 unidades.

A raíz de que buscamos que el *todo* sea la suma de las partes, el resultado anterior, debería obtenerse a partir de la suma de la cantidad de información en el caso más simple, es decir, dado que la fuente de información más rudimentaria transmite dos mensajes, entoces la cantidad de información mínima es de 2 unidades a partir de 2 símbolos.

Quiere decir entonces que cada símbolo de cualquier fuente de información debería tener una cantidad de información de 2 unidades, y si la fuente de información contiene tres símbolos posibles, entonces la cantidad de información sería de:

$$2 + 2 + 2 = 6$$

Sin embargo, nosotros observamos que la cantidad de información a partir de 3 símbolos es de 8 unidades y no de 6. Lo que deseamos hacer es convertir $2 + 2 + 2$ en 8. Aunque parece una locura, la respuesta la encontramos en la función logaritmo.

De las propiedades de los logaritmos [APO92], que no explicaremos en nuestra trabajo porque no cubre los objetivos planteados, sabemos que es posible relacionar la suma con el producto, es decir:

$$\begin{aligned} \log_2 2 + \log_2 2 + \log_2 2 &= \log_2(2)(2)(2) \\ &= \log_2 8 \end{aligned}$$

Hemos optado por la base 2 en el logaritmo, debido a que nos ayudará a simplificar los cálculos: si la fuente de información más simple transmite dos mensajes, entonces la cantidad de información es $\log_2 2 = 1$.

Para identificar que la cantidad de información ha sido determinada mediante $\log_2$, llamaremos al resultado *bit* [SIN82], de lo contrario la cantidad de información llevará el nombre de *nats*, por ejemplo, un bit de información es equivalente a $\log_{10} 2 = 0.30103$ *nats* de información.

Con esta explicación hemos definido cómo medir la cantidad de información, aunque el modelo presentado todavía tiene algunas impurezas.

El trabajo que hemos realizado para medir la cantidad de información se simplifica en encontrar el número de mensajes de una fuente de información, sin embargo, en la realidad no siempre todos los mensajes son transmitidos. Por ejemplo, en el desierto de Chihuaha el mensaje: *lloverá el dia de hoy* es algo poco probable.

Para reforzar nuestro argumento, regresando al ejemplo del timbre que tratamos en un inicio, se deduce que si éste estuviera ubicado en una casa de una colonia solitaria, entonces la probabilidad de ser activado sería menor a la obtenida cuando estuviera en una compañía de relaciones públicas en una zona transitada.

Quiere decir entonces, que a cada mensaje se le debe asociar cierta probabilidad de ser transmitido, dicha probabilidad la vincularemos con lo novedeso de un mensaje.

Por ejemplo, si la probabilidad de que el mensaje $\{0\}$ sea transmitido es de 0.9 y para el mensaje $\{1\}$ es de 0.1, la siguiente tarea es sumar la cantidad de información con la que contribuye cada mesaje.

Es decir, será necesario calcular: $(0.1 \log_2 0.1 + 0.9 \log_2 0.9) = -0.476$ bits. Por comodidad se acordó que la cantidad de información fuera una número positivo [SIN82], por lo tanto hay que interponer un signo menos en la última expresión, es decir, $-(0.1 \log_2 0.1 + 0.9 \log_2 0.9) = 0.476$

Como la probabilidad es un valor mayo que cero y menor que uno, al obervar la Figura 2.1 en donde graficamos los datos correspondientes para la función $\log_{10} x$ con $0 < x < 1.6$, observamos que para cualquier probabilidad la función logaritmo deberá ser un valor negativo.

De la misma Figura, tambien se observa que la función logaritmo en el punto cero no está definido. Sin embargo si se tiene una probabilidad igual a cero, entonces significa que no contribuye con alguna información y por lo tanto no hay que calcular $\log 0$.

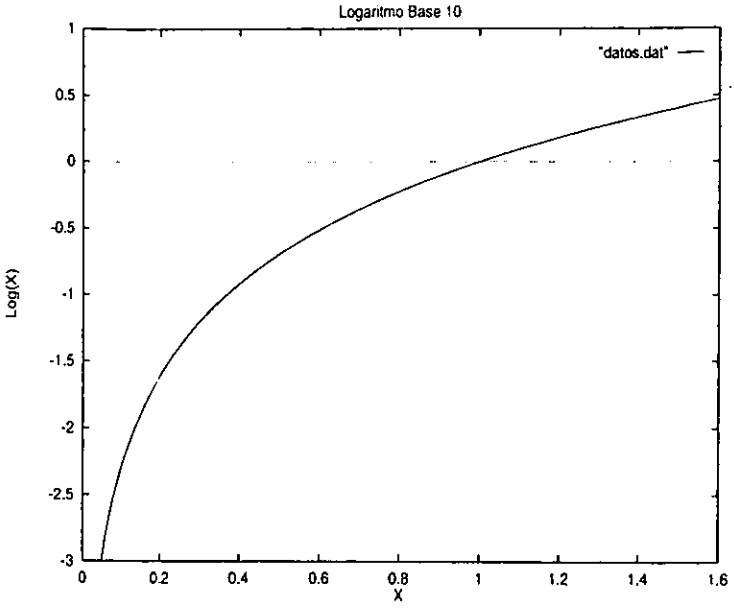


Figura 2.1: Función logaritmo base 10

Una razón matemática para decir que $\log 0$ no tiene sentido, la podemos dar muy fácilmente a partir de la definición de logaritmo:

$$\begin{aligned} \log_x y &= n \\ &\Rightarrow \quad y = x^n \end{aligned}$$

Si consideramos que $y = 0$ entonces significaría que:

$$\begin{aligned} \log_x 0 &= n \\ &\Rightarrow \quad x^n = 0 \end{aligned}$$

Lo cual es imposible, dado que no existe un número n tal que $x^n = 0$. Para una demostración más profunda y riguroza respecto al comportamiento de dicha función sugerimos [SPI92, APO92].

Del último cálculo presentado, hemos obtenido una cantidad de información de 0.47, la cual es menor a 1 bit, que antes habíamos calculado, la justificación se encuentra en la menor libertad para transmitir un mensaje, es decir, existe una mayor cantidad información cuando se comunican mensajes poco probables en vez del caso inverso.

Por ejemplo, en el desierto de Chihuha, decir *hoy hace calor* es un mesaje muy probable, dado las condiciones climáticas del estado, situación diferente si comunicamos un mensaje poco probable como *mañana lloverá*, el cual comunicaría mucha información.

Otro ejemplo muy concreto se encuentra relacionado con el amor, si algún enamorado a diario trasmite el mensaje *te quiero* a su amada, ya no será novedoso volverlo a comunicarlo, logrando con ello que una vez que ha sido escuchado durante un tiempo lo único que se sabe es que dicho mensaje es redundante, esto es equivalente a decir que la probabilidad de seleccionar el mensaje *te quiero* es muy alta, teniendo por consiguiente una disminución en la cantidad de información.

El ejemplo es muy interesante para las mentes creativas; tal es el caso de los poetas, los cuales explotan la redundancia de la lengua, de tal forma que el mensaje *te quiero* dicho mil veces se expresa de diversas maneras utilizando una variedad de mensajes poco probables, alcanzando con ello que la cantidad de información no disminuya y siga siendo novedoso para quien recibe el mensaje.

La cantidad de información se le conoce por el nombre de *Entropía*, aunque nosotros calculamos la entropía de una fuente de información con 2 mensajes, para hacer la generalización indicaremos las propiedades que deberá tener la fuente de información general.

En primer lugar requerimos una serie de símbolos que construirán nuestros mensajes: emplearemos un conjunto finito L de caracteres, por ejemplo $L = \{$ letras del alfabeto $\}$ o $L = \{0, 1\}$ por mencionar algunos casos.

También es importante considerar que la fuente de información emitirá mesaje de longitud finita N. En nuestro trabajo una fuente de información importante es el espacio de texto en claro $\mathcal{M}$, que cumple con las caracterisitas anteriores.

Es decir, si $L = \{0, 1\}$ y $N = 3$ entonces $\mathcal{M}$ está dado por;

$$\mathcal{M} = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

El número de elementos de $\mathcal{M}$ lo indentificaremos con la letra n. Dado que $\mathcal{M}$ es un conjunto finito, es factible enumerar cada uno de sus elementos, es decir:

$$\mathcal{M} = \{M_1, M_2, \ldots M_n\}$$

Con la observación anterior, de igual forma, al escribir p_1 nos referiremos a la probabilidad de transmitir el mensaje M_1, en una notación más compacta escribiremos p_i y con ello nos referiremos a la probabilidad de transmitir el mensaje i-esimo, donde el valor de i estará entre 1 y n. Por lo anterior, la entropía $H(\mathcal{M})$ se define como [MAS87]:

$$\begin{aligned} H(\mathcal{M}) &= -(p_1 \log_2 p_1 + \ldots + p_n \log_2 p_n) \\ &= -\sum_{i=1}^{n} p_i \log_2 p_i \end{aligned}$$

Para verificar que la fórmula presentada no se contrapone con nuestros resultados anteriores, verifiquemos lo que en un principio nosotros consideramos como la cantidad de información para la fuente de información $\mathcal{M}$:

$$\mathcal{M} = \{000, 001, 010, 011, 100, 101. 110, 111\}$$

En donde cada mensaje tenía la misma probabilidad de ser transmitido, es decir:

$$p_i = \tfrac{1}{n}$$

Las propiedades de nuestra fuente de información fueron de $n = 8$, $L = \{0, 1\}$, $N = 2$ y cada mensaje tenía la misma probabilidad de ser transmitido. Nosotros concluimos que la cantidad de información era $\log_2 8$ y ahora queremos verificar el resultado con la fórmula que hemos incluido.

$$\begin{aligned} H(\mathcal{M}) &= -\sum_{i=1}^{n} p_i \log_2 p_i \\ &= -\sum_{i=1}^{n} \tfrac{1}{n} \log_2 \tfrac{1}{n} \\ &= \sum_{i=1}^{n} \tfrac{1}{n} \log_2 (\tfrac{1}{n})^{-1} \\ &= n\tfrac{1}{n} \log_2 n \\ &= \log_2 n. \\ &= \log_2 8 \end{aligned}$$

Además de haber obtenido el mismo valor, también observamos que la entropía depende de la probabilidad de cada uno de los mensajes. También es importante señalar que existen fuentes de información muy diveras, por ejemplo si proponemos ahora $L = \{$ *letras del alfabeto* $\}$ y a $N = 1$ y a $\mathcal{M} = \{$ *letras del alfabeto* $\} = L$ y $n = 26$, entonces estaríamos consideramos como fuente de información a las letras del alfabeto y si suponemos además que la probabilidad de transmitir cada uno de los mensajes es igual para todos los mensajes, entonces la entropía sería de $\log_2 26$.

Para conocer la probabilidad real de cada uno de los mensajes de la fuente de información anterior, es ineludible considerar a un idioma en particular y después conocer la frecuencia relativa de cada una de las letras en varios textos y finalmente el valor encontrado será la probabilidad que estamos investigando.

Por ejemplo, para el español, después de seguir el procedimiento anterior, determinamos la distribución de frecuencias relativas para cada unas de las letras del alfabeto castellano y concluimos que no todas las letras tienen una misma frecuencia relativa.

Para ilustrar la nota anterior incluimos la Figura. 2.2, en donde la letra "E", marcada con el número 4, tiene el valor máximo, lo que implica una frecuencia relativa mayo y por lo tanto una probabilidad mayor.

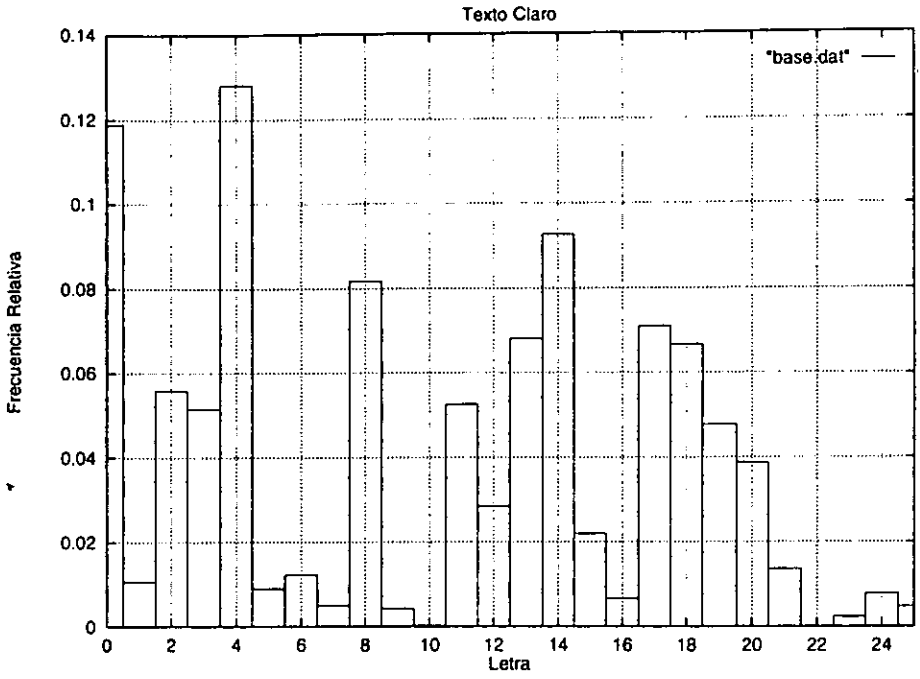


Figura 2.2: Frecuencias relativas del alfabeto en base al Español

Para saber si es relevante la frecuencia relativa de cada una de las letras, es importante cuestionar las consecuencias cuando el número de caracteres disminuye en un mensaje, por ejemplo: *Quien se fué a la villa perdió su silla*, convertido en *Quin s fu a la vlla prdi su sill.* Observamos que en el segundo mensaje no se perdió la intención del primer mensaje, aunque el segundo mensaje tiene menos caracteres que el primero.

De la observación anterior, cuando la fuente de información permite que sus mensajes sean expresados de diferente manera, sin que ello signifique una disminución en la entropía, entonces hay que meditar sobre la forma óptima para codificar un mensaje.

Si la fuente de información utiliza menos símbolos para codificar cada uno de sus mensajes y, con ello, no se disminuye la cantidad de información, entonces se ha encontrado una mejor codificación. La pregunta a resolver entonces es: ¿cuál es la codificación adecuada?.

Para dar respuesta a la pregunta anterior, analicemos el caso de $\mathcal{M} = \{A, B, C\}$, donde definiremos arbitrariamente cada una de las probabilidades por $\frac{1}{2}, \frac{1}{4}, \frac{1}{4}$ para cada mensaje, y por lo tanto, la cantidad de información estará dada por:

$$\begin{aligned} H(\mathcal{M}) &= -\sum_{i=1}^{n} p_i \log_2 p_i \\ &= -\tfrac{1}{2}\log_2 \tfrac{1}{2} - \tfrac{1}{4}\log_2 \tfrac{1}{4} - \tfrac{1}{4}\log_2 \tfrac{1}{4} \\ &= \tfrac{1}{2}\log_2 2 + \tfrac{1}{2}\log_2 4 \\ &= 0.5 + 1.0 \\ &= 1.5 \end{aligned}$$

Una codificación posible es mediante la siguiente asignación:

Mensaje	Codificación
A	00
B	10
C	11

Por lo que la serie de mensajes *A B A A C A B C* será equivalente a *00 10 00 00 11 00 10 11* en donde hemos utilizado 16 símbolos en total para transmitir 8 mensajes, quiere decir que el promedio de bits utilizados en promedio para cada mensaje es de $2 = \frac{16}{8}$.

Sin embargo por el cálculo anterior, la cantidad de información que transmite la fuente de información es de 1.5 bits, por lo que la codificación anterior no es una buena elección, utiliza más bits para decir lo mismo. Para mejorarla es necesario asignarle al mensaje que tiene una mayor probabilidad de ser comunicado un número menor de símbolos, por ejemplo:

Mensaje	Cofificación
A	0
B	10
C	11

La misma serie de mensajes *A B A A C A B C* ahora la codificaremos como *0 10 0 0 11 0 10 11*. Ahora hemos utilizado solamente 12 bits para transmitir 8 mensajes, y observamos que el promedio de bits utilizados por mensaje es de 1.5. Por lo que hemos encontrado una codificación óptima, dado que la fuente de información emite 1.5 bits de información y nosotros comunicamos el mismo número de bits.

De lo anterior, concluimos que la entropía ayudará para indicar cuándo una codificación es o no es adecuada [STA95], dado que si la codificación es óptima, entonces el promedio de bits utilizados por cada mensaje es equivalente a la entropía.

Para dar fin a la presente sección, unicamente incluiremos algunos resultados que serán importantes en la siguiente sección.

1. **Cantidad de información por elemento del mensaje (r)**: Dado que $H(\mathcal{M})$ equivale al promedio de bits utilizados en una codificación óptima y N es el número de elementos que tiene cada mensaje, entonces cada elemento de un mensaje tendrá una cantidad de información igual a $\frac{H(\mathcal{M})}{N}$.

2. **Cantidad de información de L (R)**: Los posible símbolos que conforman un mensaje se agrupan en el conjunto L. si consideramos a este conjunto como una fuente de información, la cantidad de información que transmite será de $H(L)$. En nuestro caso particular, nosotros supondremos que cada uno de los símbolos tienen la misma probabiliad de ser transmitidos, entonces $R = H(L) = \log_2 L$

3. **Redundancia (D)**: Dicho parámetro lo definimos como $R - r$.

Resumiendo, con lo que hemos trabajado hemos definido a la entropía como la cantidad de información de una fuente de información, así como el promedio de bits utilizados en una codificación óptima, al igual que hemos introducido ciertos resultados que se desprenden del mismo trabajo.

Ahora sólo nos resta aprovechar dicho esfuerzo para aplicarlo en el criptoanálisis. En la siguiente sección explicaremos cómo la *distancia de unicidad* nos relacionará todos los resultados aquí mostrados.

2.1.1 Distancia de Unicidad

Una vez introducidos los conceptos básicos de la Teoría de la Información, el siguiente paso es concretar dicho conocimiento en un resultado que determinará la cantidad de texto cifrado requerido para encontrar una solución única sin el conocimiento de la llave.

El encontrar una solución única a un texto cifrado es análogo a encontrar el texto en claro, y además, consideraremos que se desconoce el valor de la llave necesaria.

El mencionar que no existe una solución o que ésta no es única significa que al texto cifrado es imposible definirle un texto en claro o simplemente no existe la forma para determinarlo, a menos que se cuente con la correspondiente llave.

Concluir que un texto cifrado no tiene solución es un planteamiento muy interesante, reflexionemos sobre su importancia: en la sección 1.3 hablamos de las características de las funciones de cifrado y descifrado, y observamos que si dichas funciones no eran biyectivas entonces habría confusión para recuperar el texto en claro a partir de un cifrado. y por lo tanto *no existiría una solución* para el texto cifrado.

Como veremos a lo largo de la presente sección, para concluir que un texto cifrado no tiene solución, no será necesario utilizar como hipótesis que las funciones de cifrado y descifrado **no** son biyectivas. De hecho, el resultado que obtendremos no se contrapone con la definición de criptosistema.

La razón de estudiar el planteamiento relacionado con la busqueda de una solución para un texto cifrado, surge a raíz de suponer que un criptoanalista intercepta un texto cifrado con dos caracteres; con dichos elementos es imposible determinar qué texto en claro se le puede asociar; en la sección 1.8 hicimos dicha observación.

La siguiente pregunta obligada gira alrededor de la cantidad mínima que debe intereceptar el criptoanalista para garantizar que C tiene un único M asociado cuando se desconoce el valor de k.

No basta afirmar que es esencial obtener "suficiente" texto cifrado, una pregunta como: "¿cuánto es suficiente?" no tiene respuesta porque, para algunos tres caracteres es lo adecuado, mientras que para otros significan 100, por lo tanto, es primordial cuantificar la cantidad mínima evitando que el calificativo "suficiente" quede a elección personal.

Para cumplir dicha misión, utilizando los resultados de la sección anterior será muy fácil la tarea. En primer lugar necesitaremos contar el número de llaves que existen en $\mathcal{K}$.

Sabemos que $H(\mathcal{K})$ equivale al número promedio de bits utilizados en una codificación óptima. Si conocemos el promedio de bits que se utilizarán, el siguiente paso será contar el número posible de combinaciones que se pueden realizar con ese número de bits.

Por ejemplo, observamos que al transmitir 3 bits, el número de combinaciones es de 2^8, de hecho, de manera genérica, el número de combinaciones a partir de n bits, es de 2^n.

Aunque no conocemos el número exacto de bits empleados, conocemos el *promedio* de bits utilizados, el cual será suficiente, y por lo tanto el número posible de llaves estará dado por;

$$\text{Número de llaves posibles} \qquad 2^{H(\mathcal{M})}$$

Dentro de todas las posibles llaves, existirá una llave que es la que el criptoanalista desconoce y precisamente quiere encontrar, por lo que, a excepción de una sola llave, el resto de las llaves no servirán para descifrar un mensaje particular, es decir:

Número de llaves:		
en $\mathcal{K}$	$=$	$2^{H(\mathcal{K})}$
correctas para descifrar un texto particular	$=$	1
incorrectas para descifrar un texto particular	$=$	$2^{H(\mathcal{K})} - 1$

Por otra parte, si ahora contamos los elementos que existen en $\mathcal{M}$, dado que los mensajes son de longitud N finita, análogamente el número de mensajes posibles es de $2^{H(\mathcal{M})}$, y escribiendo de diferente forma el resultado tenemos que:

$$\begin{aligned} \text{Número de mensajes en } \mathcal{M} &= 2^{H(\mathcal{M})} \\ &= 2^{H(\mathcal{M})\frac{N}{N}} \\ &= 2^{\frac{H(\mathcal{M})}{N}N} \\ &= 2^{rN} \end{aligned}$$

Tanto para contar el número de elementos de $\mathcal{K}$ y $\mathcal{M}$ hay que puntualizar que son aproximaciones que estamos realizando, dado que queremos aprovechar el resultado de la entropía en vez de enumerar el número exacto de elementos que forman cada uno de estos conjuntos.

Dado que $H(\mathcal{M})$ está en función de las probabilidades de cada uno de sus mensajes de un idioma, donde cada mensaje tendrá una interpretación particular, ahora supondremos el caso donde cada mensaje tiene la misma probabilidad de ser transmitido, y por ende se tendrán mensajes que no tienen sentido.

Dado que también deseamos contar el número de elementos que existen en este conjunto, consideremos que cada símbolo empleado en la construcción de un mensaje tiene la misma probabilidad de selección, por lo tanto el número de bits en promedio utilizados es de:

$$\begin{aligned} H(L) &= \log_2 L \\ &= R \end{aligned}$$

Ahora si queremos construir un mensaje con N bits, entonces el mesaje tendrá un tamaño de $(R)(N)$. Por lo tanto, el número de mensajes posibles es de:

$$\text{Número de mensajes posibles} = 2^{RN}$$

Con los resultados anteriores, calculemos ahora a probabilidad P de obtener un mensaje con sentido, es decir, como P es el cociente del número de mensajes que tienen sentido y el número de mesajes posibles, la expresión de P es:

$$\begin{aligned} P &= \frac{2^{rN}}{2^{RN}} \\ &= 2^{rN-RN} \\ &= 2^{N(-D)} \\ &= 2^{-ND} \end{aligned}$$

A excepción de una llave, el resto de llaves probadas en la función de descifrado dará por resultado mensajes que son distintos al buscado, entonces, una aproximación de la probabilidad para obtener un texto que no es el deseado es P.

Y si consideramos el producto entre el número de llaves que no darán por resultado un texto en claro correcto, y la probabilidad de obtener un mensaje sin sentido, obtenemos así el número esperado de soluciones falsas, es decir:

$$\begin{aligned} (2^{H(K)} - 1)2^{-DN} & \\ &= 2^{H(K)-DN} - 2^{-DN} \\ &\approx 2^{H(K)-DN} \end{aligned}$$

A la expresión anterior llamémosla F y observemos que hemos despreciado la contribución de 2^{-DN}, ya que el aumento de N implica que el cociente $2^{-DN} = \frac{1}{2^{DN}}$ se aproxima a cero.

Ahora, apliquemos la función logaritmo base dos en ambas partes de la expresión F para obtener:

$$\begin{aligned} \log_2 F &= \log_2 2^{H(\mathcal{K})-DN} \\ &= H(\mathcal{K}) - DN \end{aligned}$$

En el momento en el que el número esperado de falsas soluciones es muy pequeño entonces es factible inferir el texto en claro, es mas, para ser mas claro, en el momento de que no existe una solución falsa entonces se ha encontrado la solución correcta al texto cifrado dado. es decir, nos interesa el caso cuando:

$$\begin{aligned} \log_2 F &= H(\mathcal{K}) - DN \\ &= 0 \\ &\Rightarrow H(\mathcal{K}) = DN \\ &\Rightarrow N = \tfrac{H(\mathcal{K})}{D} \end{aligned}$$

Por lo tanto, de la expresión anterior se observa que es suficiente obtener $N = \frac{H(\mathcal{K})}{D}$ caracteres para encontrar una solución al texto cifrado. El resultado anterior lleva el nombre de *distancia de unicidad* [SIN66].

Entonces, para estudiar el caso cuando no existe una solución a un texto cifrado es necesario suponer como hipótesis que el número de llaves es tan grande como el número de mensajes posibles.

En la sección 2.1 concluimos que si una fuente de información L tiene n mensajes con la misma probabilidad de ser transmitido, entonces $H(L) = \log_2 n$. Dado que cada elemento de $\mathcal{K}$ tiene la misma probabilidad de selección, entonces solo nos resta conocer el número de elementos que tiene $\mathcal{K}$.

Como hemos deseado que el número de llaves sea tan grando como el de mesajes, entonces necesitaremos conocer el número de mensajes posibles que se pueden formar con N caracteres. Como indicamos al inicio de esta sección, el número de mensajes posibles es:

$$2^{RN}$$

Por lo tanto el valor $H(\mathcal{K})$ se define como:

$$\begin{aligned} H(\mathcal{K}) &= \log_2 2^{RN} \\ &= RN \end{aligned}$$

Sustituyendo dicho valor en la fórmula de la distancia de unicidad, tenemos que:

$$\begin{aligned} N &= \frac{H(\mathcal{K})}{D} \\ &\Rightarrow DN = RN \\ &\Rightarrow DN - RN = 0 \\ &\Rightarrow N(D-R) = 0 \\ &\Rightarrow R - r - R = 0 \\ &\Rightarrow r = 0 \\ &\Rightarrow \frac{H(\mathcal{M})}{N} = 0 \\ &\Rightarrow H(\mathcal{M}) = 0 \end{aligned}$$

Quiere decir que el número de bits en promedio para codificar un mensaje dado es de **cero** bits, por lo tanto no existe información en el mensaje, lo que se traduce como no poder encontrar una solución al texto cifrado.

El resultado es interesante, ahora hemos establecido una condición suficiente para garantizar que un texto cifrado sea indecifrable a menos que se cuente con la llave adecuada [ROB82], sin necesidad de modificar las propiedades de las funciones de cifrado y de descifrado.

Hay que observar que el hecho de que exista la solución al texto cifrado no significa que sea fácil o que requiera de pocos recursos de cómputo para encontrarla, lo único que se asegura es que si se intercepta menos texto cifrado del indicado por la distancia de unicidad, entonces no existirá una solución al texto cifrado, puesto que el número esperado de soluciones falsas jamás se aproxima a cero.

También hay que puntualizar que la distancia de unicidad no se aplica en el caso de la criptografía de llave pública, dado que la tarea principal del criptoanalista es la de poder inferir la llave privada a partir de la pública y por lo tanto no utilizará cierta cantidad de texto cifrado.

Para finalizar la presente sección queremos enfatizar que gracias a los conceptos estudiados se establece la condición suficiente para obtener un mensaje 100 por ciento seguro. En la sección 3.1 mostraremos un ejemplo de lo afirmado y aunque a través de la historia se intercambiaron mensajes con dicha propiedad, se desconocían las matemáticas involucradas [KAH67] que hemos abordado ahora.

A pesar del resultado, que indica la forma de obtener un texto indescifrable, el reto actual se basa en establecer funciones de cifrado "casi" indescifrables sin utilizar la condición que hemos estudiado.

Es importante mencionar que el mérito de los resultados que hemos explicado fueron obtenidos en 1945 en los estudios realizados por Shannon [ROB82], los cuales dieron los fundamentos teóricos para la criptografía, precisamente lo que hemos mostrado son los elementos básicos que Shannon introdujo.

2.2 Teoría de Números

Cuando iniciamos nuestros trabajos escolares relacionados con los números. seguramente la etapa inicial consistió en utilizarlos en operaciones aritméticas: dicha misión se redujo a obtener "habilidad" en su manejo y difícilmente nos acercamos a una verdadera comprensión de ellos.

En esta sección nos vamos a limitar al estudio de algunas propiedades muy simples de los números enteros, cuya área de las matemáticas llamada *Teoría de Números* es la encargada de estudiarlos. Con ello esperamos dar un paso hacia una segunda etapa en el estudio de los números, que reside en aproximarnos a su verdadero potencial.

Cualquier estudiante desde nivel bachillerato sabe que el conjunto de los números enteros está formado por el conjunto $\mathbb{Z}$ definido como:

$$\mathbb{Z} = \{\ldots, -4, -3, -2, -1, 0, 1, 2, 3, 4\ldots\}$$

y además conoce la forma de operar con ellos muy bien, veremos posteriormente que existe un fondo más complejo del que nos imaginamos.

El camino en el estudio de los números enteros, eventualmente es extremadamente laborioso; nos tropezamos con cuestionamientos sobre la *existencia* de éstos o reflexiones sobre su estructura, por mencionar algunas.

En el estudio de los números enteros hay que mencionar que existen muchas interrogantes, que son retos para su explicación y comprensión, y por supuesto, de gran interés para las matemáticas: un estudio más detallado respecto a tópicos especializados se encuentran en [FRA87, WIJ77, HAR92, APO92, SPI92, CAR90].

Uno de los sustanciales conflictos con campos fundamentalmente téoricos, como lo es la Teoría de Números, es que en ocasiones se consideran inútiles, antes de explicar la razón por la cual hemos incluido esta sección, es importante reflecionar en algunos sucesos históricos.

Contemos una curiosa anécdota [HAR92] que tuvo lugar en Inglaterra, donde un estudiante Indú, llamado Ramanujan, cuyo profesor era un gran matemático, sufrió de una enfermedad que lo llevó al hospital. Su profesor, preocupado por su salud fué a visitarlo; durante su trayecto se subió a un taxi cuyo número de placas era 1729. El profesor, quien no le dió importancia, le comentó dicho número a su alumno.

Al parecer no hay algo relevante en la narración anterior, ya que tanto para el profesor como para nosotros dicho número no significa nada más que el número que identifica un vehículo, sin embargo, Ramanujan observó que dicho número es el menor entero positivo que se puede expresar como la suma de dos cubos distintos, es decir:

$$\begin{aligned} 1729 &= 10^3 + 9^3 \\ &= 12^3 + 1^3. \end{aligned}$$

Resultados tan extraños para nosotros hacen la diferencia entre *manejar* y *comprender* a los números enteros, y veremos cómo al enfrentarnos con planteamientos utilizando hipótesis simples resultarán una fuente de sabiduría.

Muchos de los *Teoremas* o verdades demostradas parecen no tener una aplicación en el mundo real por la complejidad que representan; sin embargo, en la actualidad la criptología basa su funcionamiento en un sinfín de resultados de esta naturaleza y es por ello que aquellos matemáticos vistos como "no aplicados" encuentran ahora la oportunidad de contribuir de inmediato a la sociedad proponiendo mejores funciones de cifrado.

El interés por la Teoría de Números no es nuevo, las aportaciones emanaron de una infinidad de matemáticos: uno de los primeros esfuerzos comenzaron con Euclides con la consumación de su serie de libros llamados: *Elementos*, en donde se dieron las bases tanto para la Teoría de Números como para la geometría.

Un personaje muy famoso en la Teoría de números se llama Pier Fermat [1601-1665], cuyas observaciones, encontradas a partir de la comunicación escrita con otro gran matemático de nombre Mersen, provocó en los matemáticos contemporáneos grandes dolores de cabeza [HAR92].

Lo más impresionante de este personaje, es que Fermat no era matemático de profesión, pero su pasión por las matemáticas lo llevó a proponer y cuestionarse sobre razonamientos inimaginables; aunque en muchas ocasiones no anexaba las demostraciones.

Una de las conjeturas expuestas por él, que durante mucho tiempo no fue demostrada, es aquella, mal llamada *Último Teorema de Fermat* y que no fué si no hasta mediados de la década de los 90 cuando por fin se demostró.

El planteamiento está relacionado con una ecuación cuya solución no es posible en los números enteros, es decir, no existen enteros x, y, z tales que:

$$x^n + y^n = z^n$$

No tiene solución en los enteros para $n > 2$.

En nuestro trabajo no abordaremos enigmas tan obscuros, lo mencionamos por la fama que tomó y es común leer un libro de Teoría de Números y observar algún comentario referente a dicha conjetura.

Contemporáneo a Fermat, también otros de los grandes precursores tanto en la Teoría de Números, como de una variedad de otras áreas de las matemáticas, fue Carl Friedrich Gauss [1787-1855], quien desarrolló el lenguaje de congruencias [ROS93], que explicaremos en la sección 2.2.2, así como también proporcionó las bases para la Teoría de Números moderna en su libro *Disquistiones Arithmeticae* en 1801. Por el extenso material con el que contribuyó a la humanidad se le ha considerado el matemático más importante de la historia.

Lo que nos enseñaron estos matemáticos sirvió para recapacitar sobre la inmensa vista alrededor de los números enteros, la cual sobrepasa las fronteras de simples sumas o restas. De igual forma, debido a su gran aplicación en la criptografía, a retomado mucho más interés que antes.

A pesar de que la Teoría de Números contempla también a la aritmética, su campo es ambicioso y evidencia que los números enteros ofrecen más conclusiones que "1+1=2". Sin embargo, a partir de planteamientos muy simples se llegan a conclusiones sumamente complicadas.

La aspiración principal de la actual sección es la de esclarecer las ideas básicas de la Teoría de Números, las cuales son las adecuadas para cubrir los objetivos generales de nuestro trabajo. pero sin duda también esperamos incorporar conceptos más avanzados con la finalidad de proceder hacia la comprensión de la criptografía moderna.

Como comentario final, antes de iniciar nuestro recorrido, hay que mencionar que el material aquí presentado significó uno de los esfuerzos más grandes para su redacción, puesto que en la diversa literatura comúnmente uno se enfrenta con frases como: "es obvio que...", "fácilmente se demuestra que...".

Por lo tanto, después de comprender los resultados analizados, es cuando uno entiende y comparte la frase dicha por Nathaniel Bowditch [1773-1838] cuando tradujo el libro *Mécanique céleste* de Laplace [WIL90]: *No puedo encontrar una afirmación de Laplace "así es evidente", sin tener la seguridad de que deberé emplear horas de trabajo intenso. para cubrir el abismo y averigurar y demostrar lo evidente que es.*

Es por ello que parte de nuestra motivación en esta sección radica en cubrir abismos de "trivialidades" a las que uno se enfrenta en la diversa literatura [ROS93, HAR92, WIJ77, ROB82], de tal forma que queden claros los postulados y se cuenten con las bases para seguir profundizando en los temas que se tengan interés.

2.2.1 Divisibilidad

El lenguaje es uno de los principales factores que facilitan la transmisión del conocimiento; las matemáticas tienen el suyo el cual adquirimos a través de la escuela. A continuación vamos a partir de una operación muy difundida: la división y desarrollaremos un nuevo lenguaje asocido a ésta.

En ocasiones para calcular una división utilizamos lo que conocemos como "casita" (dicho nombre no lo encontramos justificado en alguna bibliografía): por ejemplo, si queremos dividir 12 entre 6, realizamos la operación como en la Figura 2.3, donde ponemos adentro al 12 y afuera al 6.

$$\begin{array}{r|l} & 2 \\ \hline 6 & 12 \\ & 0 \end{array}$$

Figura 2.3: División de dos números

La notación que utilizamos para representar la operación elaborada en la Figura 2.3 es mediante el cociente:

$$\frac{12}{6} = 2$$

Claro es que en algunas ocasiones la operación se realiza mentalmente y los maestros que nos instruyeron en dicho cálculo nos preguntan: "¿cuántas veces cabe el 6 en el 12 ?" y con ello damos una respuesta acertada.

De la experiencia con los números, aprendimos que no siempre al dividir un número entre otro tenemos por resultado un entero. por ejemplo $\frac{5}{7}$ no es un entero, e inclusive utilizamos dicha operación con otro tipo de números; una muestra de ello, a pesar del error numérico, es $\frac{\pi}{9}$.

La razón por la cual hacemos esta reflexión previa es para ilustrar que empleamos otras clases de números, y al operar con ellos nunca nos cuestionamos sobre el tipo de números empleados. Sin embargo eso no siempre fue así: por ejemplo, en una comunidad donde exclusivamente tuvieran conocimiento de los números enteros y les plantearamos resolver la ecuación $2x = 5$ seguramente no sabrían que hacer; la humanidad a través de su evolución estuvo en una situación similar.

Es importante que olvidemos un poco de nuestros hábitos con los otros números y nos concentremos en pertenecer a este grupo que únicamente está familiarizado con los números enteros; para facilitar el proceso de olvido es necesario identificar lo que debemos desechar.

A partir de la simbología utilizada para realizar una división, supongamos que queremos conocer el valor al dividir dos enteros a y b, cuya respuesta la encontramos en la Figura 2.4.

El resultado final lo escibimos como $\frac{b}{a} = q\frac{r}{a}$, por ejemplo $\frac{3}{2}$ lo expresamos como $1\frac{1}{2}$ así como no es de extrañarse que $\frac{5}{3} = 1\frac{2}{3}$. Al número q lo conocemos por el nombre de *cociente* y a r como *residuo*.

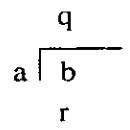


Figura 2.4: División de b entre a

La forma correcta de expresar el resultado de la Figura 2.4 es:

$$\frac{b}{a} = q + \frac{r}{a},$$

Si focalizamos la atención en la última expresión, deducimos que al multiplicar ambos lados por a llegamos a:

$$b = aq + r.$$

En donde el residuo r tendrá valores mayores o iguales que cero, y menores que a, y q es cualquier entero.

Ahora sí estamos listos para formalizar un criterio de división entre los número enteros; en primer lugar necesitamos que el residuo sea igual a cero ($r = 0$) y para no confundir el término cocientes que hemos trabajado por un largo tiempo es indispensable una definición de divisibilidad en la que no se haga referencia al concepto anterior, para hacerlo utilizamos la siguiente [CAR90]:

Sean b y $a \neq 0$ dos números enteros, decimos que a divide a b si existe otro entero q tal que $b = aq$ y lo denotaremos como:

$$a|b$$
$$a \text{ divide a } b$$

En caso contrario escribimos $a \nmid b$, es decir que a no divide a b.

Con la notación señalada, es más claro por qué la división por cero no tiene sentido ya que de ser admisible entonces: $0|b \Rightarrow b = 0q$, pero no existe algún q de tal forma que $q0 = b$, con $b \neq 0$.

Utilizando el concepto de divisibilidad entre dos enteros, ahora vamos a estudiar un concepto que necesitaremos en un futuro: *Máximo común divisor*

Máximo Común Divisor

El cuestionamiento que da origen al máximo común divisor se relaciona con los divisores de un número; por ejemplo, si tomamos el entero 24, mediante ensayo y error encontramos cuáles son aquellos enteros que dividen a 24; de ahí concluímos que dicho conjunto se define como:

$$A = \{-24, -12, -6, -4, -3, -2, -1, 1, 2. 3, 4, 6. 12, 24\}$$

Si partimos de un entero menor la labor de ensayo y error es más simple, por decir el 18, vemos que sus divisores son:

$$B = \{-18, -9, -6, -3, -2, -1, 1, 2, 3, 6, 9, 18\},$$

Ya que hemos empleado éstos dos enteros, intuimos que tienen en *común* algunos divisores; para denotar los elementos comunes entre dos conjuntos utilizaremos el símbolo $\bigcap$, con ello $A \bigcap B$ significa el conjunto que contempla los elementos que se encuentran tanto en A como en B, en nuestro caso particular tenemos:

$$A \bigcap B = \{-6, -3, -2, -1, 1, 2. 3, 6\}$$

Del conjunto anterior se observa que el máximo de ellos es el 6 y además éste es dividido por los otros divisores comunes [WIJ77]. Para identificarlo lo llamaremos *Máximo común divisor.*

Para encontrar el máximo común divisor ya hemos propuesto un método: mediante ensayo y error encontramos los números que dividen a a, después los que dividen a b, escogemos los divisores comunes e identificamos el número mayor del conjunto.

Para identificar el máximo común divisor de dos enteros a y b emplearemos como notación: $d = (a, b)$.

Por desgracia aquí es necesario ser precavidos para no confundir un par ordenado con el máximo común divisor; aunque se use la misma notación su significado es diferente, por lo tanto tenemos que ubicarnos en el contexto particular, afortunadamente en nuestro trabajo ya no hablamos de pares ordenados, sólo se tiene que recordar que es el del máximo común divisor.

De la definición de divisibilidad ya establecida concluímos que la unidad o el entero $a = 1$ divide a cualquier número, es decir:

$$1|b \Rightarrow b = 1q$$
$$\text{donde } q = b$$

Por lo tanto, cuando determinemos el máximo común divisor tendremos por lo menos a la unidad, por ejemplo: los divisores de 14 son $\{-14, -7, -2, -1, 1, 2, 7, 14\}$ y los de 11 son $\{-11, -1, 1, 11\}$. Como vemos $(14, 11) = 1$.

Cuando dos enteros tienen como máximo común divisor a la unidad. decimos que ambos números son *primos relativos.*

De modo más general, si desde un principio nos percatamos que los únicos divisores de un entero mayor que 1 son: $-1, +1$ y él mismo. entonces lo llamaremos *primo* y son precesiamente éstos los que toman un gran interés en nuestro estudio al igual que en la Teoría de Números.

Para mostrar el gran papel que juegan los números primos. sólo basta mencionar que todo entero mayor que +1 y -1 se puede expresar como producto de números primos. Dicha verdad lleva el nombre de *Teorema fundamental de la aritmética* [CAR90] y lo que afirma es que los números primos son el esqueleto de los números enteros.

Para ilustrar lo afirmado; por ejemplo, el número 6 se expresa como el producto de (3)(2), $10 = (5)(2)$, $8 = (2)(2)(2) = 2^3$ y no existe otra representación para hacerlo.

Gracias al Teorema fundamental de la aritmética estableceremos otro método para determinar el máximo común divisor: como cada entero se encuentra conformado por números primos entonces el máximo común divisor asimismo se encontrará compuesto de números primos. Para comprender el resultado expresemos al entero a como productos de primos:

$$a = (P_1)(P_2)\dots(P_r)$$

Quiere decir que a está formado de r primeros. Pero en la factorización a veces aparece un mismo primo varias veces, por ejemplo:

$$\begin{aligned} 24 &= (2)(2)(2)(3) \\ &= (2^3)(3^1) \end{aligned}$$

Con la consideración anterior, nosotros expresaremos al entero a como producto de primeros distintos, y en el caso de que se repita entonces escribiremos el primo con su correspondiente exponente. Es decir:

$$a = P_1{}^{e_1} P_2{}^{e_2} \dots P_r{}^{e_r}$$

Donde $P_1{}^{e_1}$ indica el primo 1 con su exponente e_1. Similarmente, al entero b lo escribiremos como producto de primos, es decir:

$$b = P_1{}^{E_1} P_2{}^{E_2} \dots P_r{}^{E_r}$$

Hay que observar que el número de primeros incluidos en a y en b son los mismos. ¿ quiere decir que $a = b$?. El usar los mismos números primos no es un error, lo hacemos por conveniencia. y para justificar que es correcto el razonamiento, consideremos que hay algún $P_i{}^{e_i}$ que está en la factorización de a y no está en la de b; para solucionar ese problema en la facotorización de b, el exponente E_i, que corresponde al primo P_i de la factorización de a, será igual a cero. Por ejemplo:

$$\begin{aligned} 12 &= (2^2)(5^0)(3)(7^0) \\ 105 &= (2^0)(5)(3)(7) \end{aligned}$$

Ahora vamos a denotar a $h_i = min\{e_i, E_i\}$, es decir el exponente h_i es igual al entero mínimo entre e_i, E_i, por ejemplo: $2 = min\{5, 2\}$.

Ya que hemos explicado todos los detalles de la notación, el máximo común divisor deberá contener a los primos que se encuentran tanto en a como en b, ya que es un divisor común, pero para garantizar que divida a ambos enteros, la potencia del primo debe ser la menor entre la potencia de ese mismo primo entre a y b.

Por ejemplo, si un factor primo de a es 5^2 y de b es 5^1. El máximo común divisor debe de contener al primo 5, pero si tomamos la potencia mayor, 2, observamos que el máximo común dividira a a, pero no dividirá a b, puesto que $25 \not| 5$.

Con fundamento en lo anterior, el máximo común divisor se define como:

$$(a, b) = P_1{}^{h_1} P_2{}^{h_2} \dots P_h{}^{h_r}$$
$$\text{con } h_i = min\{e_i, E_i\}$$

Por ejemplo, sea $a = 24$ y $b = 18$, entonces:

$$\begin{aligned} 24 &= (2^3)(3^1) \\ 18 &= (2^1)(3^2) \end{aligned}$$

De donde observamos que:

$$\begin{aligned} P_1 &= 2 \\ P_2 &= 3 \\ h_1 &= min\{3, 1\} \\ &= 1 \\ h_2 &= min\{2, 1\} \\ &= 2 \end{aligned}$$

Por lo tanto, el máximo común divisor de 24 y 18 es:

$$\begin{aligned} d &= (24, 18) \\ &= (2^{min\{3,1\}})(3^{min\{3,1\}}) \\ &= (2^1)(3^1) \\ &= 6 \end{aligned}$$

El último punto por mencionar, necesario para la sección 2.2.3, es :

$$\text{Si } (a,b) = 1 \text{ y } (c,b) = 1 \Rightarrow (ac,b) = 1$$

Para comprender la propuesta, manejando a los números primos es inmediato: puesto que no existe ningún primo común entre a y b y no lo existe tampoco entre c y b, entonces al realizar el produco entre a y c tampoco tendremos un primo que compartan con b y por lo tanto siguen siendo primos relativos.

A partir de lo que hemos desarrollado seremos capaces de construir la infraestructura que nos ayudará a ilustrar algunas funciones criptográficas elementales.

2.2.2 Aritmética Modular

La aritmética modular es equivalente a desarrollar el lenguaje de las *congruencias*, que se entienden a partir del concepto de divisibilidad, que hemos discutido en la sección 2.2.1. Nuestra misión en las siguientes líneas es la de extender el uso de la divisibilidad en todos los enteros para definir el concepto de *congruencia*.

Sabemos que al efectuar una división no siempre da como resultado un número entero, lo que significa que existe un residuo diferente de cero.

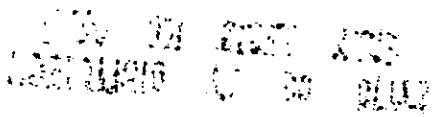
Si consideramos un par de entero a y b diferentes, y los dividimos entre m, en la sección anterior acordamos que la forma correcta de representar el resultado es mediante la siguiente expresión:

$$\begin{aligned} a &= m(q) + r \\ b &= m(q_1) + r_1 \end{aligned}$$

Para nuestro trabajo nos interesa el caso cuando $r = r_1$, es decir, los enteros a y b tienen por residuo al mismo entero r cuando son divididos por otro entero m. Hay que observar que los enteros q y q_1, que llamamos cociente, no son iguales, ya que si $q = q_1$ entonces $a = b$, pero nosotros consideramos que $a \neq b$.

De las expresiones anteriores calculemos ahora la diferencia entre a y b, es decir:

$$\begin{aligned} a - b &= mq + r - mq_1 - r \\ &= mq - mq_1 + 0 \\ &= m(q - q_1) \\ &\Rightarrow m|(a - b) \end{aligned}$$



Podemos concluir entonces, que m divide la diferencia entre dos enteros que comparten un mismo residuo. Dicho residuo se obtuvo cuando cada uno de los enteros fué divido por el entero m. En otras palabras; cuando dos enteros a y b **no** son divididos por m, pero tienen el mismo residuo, entonces la diferencia entre ambos **sí** es divida por m. Para denotar dicho resultado escribimos:

$$a \equiv b \bmod m$$
$$a \text{ es congruente } b \text{ módulo } m$$

Precisamente hemos explicado ya la definición de una congruencias, pero para ser más claro incluiremos la definición formal:

Definición de Congruencia:
Dado dos enteros a y b, a es congruente b módulo m si y solamente si m divide a $a - b$

De la definición de divisibilidad, incluida en la sección 2.2.1, mencionamos que al dividir un entero a entre m, el valor del residuo tomaría valores mayores que cero y necesariamente menor a m. Cada uno de los residuos los podemos agrupar en un conjunto que llamaremos *Sistema Completo de Residuos módulo m* (SCR), en otras palabras, SCR está formado por todos los posibles residuos cuando m divide a un entero cualquiera, es decir:

$$SCR = \{0, 1, \ldots m - 1\}$$

La primera aceveración en relación al conjunto SCR indica que cada entero será congruente con solamente con un elemento del SCR. Para proporcionar una justificación; como todo entero a no tiene dos residuos diferentes cuando es dividido por m, inferimos que si m no divide a a, entonces:

$$\begin{aligned} a &= mq + r \\ &\Rightarrow a - r = mq \\ &\Rightarrow m|(a - r) \\ &\Rightarrow a \equiv r \bmod m \end{aligned}$$

Es decir, cada entero es congruente con su residuo, y como todos los residuos posibles se encuentran en el conjunto SCR, entonces todos los enteros serán congruentes con un elemento del conjunto SCR.

En el lenguaje de las congruencias no es fundamental preguntarse si un número sea igual a otro, sino que compartan el mismo residuo. Por ejemplo, si $m = 7$:

$$7 \equiv 0 \bmod 7$$
$$14 \equiv 0 \bmod 7$$

Aunque $7 \neq 14$ vemos que: $7|(14-7)$ y por lo tanto $14 \equiv 7 \bmod 7$. Inclusive, de manera general, para cualquier entero q, $7 \equiv 7q \bmod 7$, por ejemplo, $7 \equiv 7 \bmod 7$, $7 \equiv 14 \bmod 7$, $7 \equiv 21 \bmod 7$.

Si sabemos que cada entero es congruente con un residuo del SCR, para encontrar el respectivo residuo una vez que dividimos a entre m emplearemos como notación $r = a \bmod m$, la cual es muy difundida en la jerga computacional.

Dicho de un modo formal: cualquier entero se puede reducir módulo m, y con ello encontrar el residuo positivo mínimo que se encuentre dentro del conjunto SCR.

Con el lenguaje de las congruencias ahora podremos explicar una aplicación muy concreta en el campo de la criptografía y con ello observar la potencialidad que existe con el uso del lenguaje de las congruencias.

Función de Cifrado mediante Suma módulo m

En la sección 1.5.2 ejemplificamos un criptosistema de llave secreta, en donde proporcionamos la función de descifrado y cifrado mediante pares ordenados cuando $k = 3$. Sin embargo no mostramos la expresión para E_k y D_k.

Hicimos la observación de que la finalidad de la función de cifrado era la de aplicar un corrimiento a las letras del alfabeto tres unidades hacia la derecha; es decir la letra "A" la ciframos con la tercera letra que se encuentra a su derecha, es decir, la letra "C".

Gracias a los conceptos desarrollados hasta el momento, contamos con los argumentos necesarios para justificar una expresión matemática que describa a la función de cifrado y descifrado. Para lograrlo, en una primera etapa es necesario y así lo haremos a lo largo del trabajo, asignarle a cada letra un entero. supongamos el siguiente alfabeto:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Con esta nueva asignación consideraremos a una letra como un entero. Si ahora a cada entero le sumamos otro, digamos el 3, módulo 26, conseguimos una nueva correspondencia:

0	1	2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13	14	15
13	14	15	16	17	18	19	20	21	22	23	24	25
16	17	18	19	20	21	22	23	24	25	0	1	2

El resultado es un corrimiento del alfabeto tres unidades; en nuestro ejemplo de llave secreta de la sección 1.5.2 no mostramos la expresión que realizaba dicho trabajo. Análogamente, si ahora a cada entero le restamos el entero 3 módulo 26, obtendremos la primera tabla.

El entero que sumamos corresponde a la llave escogida, por ejemplo, nosotros ilustramos el caso para $k = 3$. Sin embargo no necesariamente nos limitaremos a esta situación y por ello analicemos el caso para cualquier k.

Como el cifrado y descifrado se efectua letra por letra, nuestra función de cifrado y descifrado deberán considerar los siguientes conjuntos:

$$M = \{m_1, m_2, \ldots, m_i, m_{i+1}, \ldots, m_{N-1}, m_N\}$$
$$C = \{c_1, c_2, \ldots, c_i, c_{i+1}, \ldots, c_{N-1}, c_N\}$$

Y cada uno de los textos se obtendrán al aplicar la correspondiente función (de cifrado o descifrado) a cada uno de los elementos del conjunto, es decir, cada elemento del texto en claro o cifrádo quederará determinado del siguiente modo:

$$\begin{array}{rcll} c_i & = & E_k(m_i) = m_i + k \bmod m & 1 \leq \ i \ \leq N \\ m_i & = & D_k(c_i) = c_i - k \bmod m & \end{array}$$

En donde k, que es la llave, corresponderá al corrimiento deseado, mientras que el módulo m está sujeto al número de elementos que contiene el alfabeto que construye cada mensaje de $\mathcal{M}$. En nuestro trabajo, en una primera etapa $m = 26$ y posteriormente $m = 27$. en donde consideraremos una letra más: "ñ", y con ello ilustraremos que es posible cambiar el módulo.

Con la expresión matemática que incorporamos, utilizando $k = 3$ cifremos el mensaje $M =$ { SIN MATEMATICAS SE SUFRE MAS }

$$\begin{array}{rcl} C & = & \{E_3(\text{ SIN MATEMATICAS SE SUFRE MAS })\} \\ & = & \{E_3(m_1), E_3(m_2), \ldots, E_3(m_{22}), E_3(m_{23}), E_3(m_{24})\} \\ & = & \{m_1 + 3 \bmod 26, \ldots, m_{24} + k \bmod m\} \\ & = & \{18 + 3 \bmod 26, \ldots, 18 + 3 \bmod 26\} \\ & = & \{21, 11, 16, \ldots, 15, 3, 21\} \\ & = & \{c_1, c_2, \ldots, c_{22}, c_{23}, c_{24}\} \\ & = & \{\text{ VLQ PDWHPDWLFDV VH VXIUH PDV }\} \end{array}$$

Recordemos que el receptor también conoce la llave, no olvidemos que estamos trabajando con el esquema de llave secreta, por lo tanto para descifrar se sigue que:

$$\begin{aligned} M &= \{D_3(\text{ VLQ PDWHPDWLFDV VH VXIUH PDV })\} \\ &= \{D_3(c_1), D_3(c_2), \ldots, D_3(c_{22}), D_3(c_{23}), D_3(c_{24})\} \\ &= \{c_1 - 3 \bmod 26, \ldots, c_{24} - 3 \bmod 26\} \\ &= \{21 - 3 \bmod 26, \ldots, 21 - 3 \bmod 26\} \\ &= \{18, 8, 13, \ldots, 12, 0, 18\} \\ &= \{m_1, m_2, \ldots, m_{22}, m_{23}, m_{24}\} \\ &= \{\text{ SIN MATEMATICAS SE SUFRE MAS }\} \end{aligned}$$

Hemos alterado el texto con un mecanismo sencillo y por supuesto en la actualidad es evidente encontrar la solución a un criptosistema que empléara dicha función; solamente deberíamos intentar los 26 corrimientos diferentes, tarea que en una computadora se concluye rápidamente.

La función que hemos ilustrado, con $k = 3$ en la época del Imperio Romano fue usada por Julio César [KAH67]; Dado que mucha gente no sabía leer en aquellos días, la seguridad que proporcionaba era suficiente.

Gracias al uso de las congruencias no recurriremos al conjunto de pares ordenados para conocer la correspondencia entre el texto en claro y el cifrado. Ahora, sistemáticamente mediante la correspondencia de cada letra a un entero, calculamos la suma de k módulo m.

Sin matemáticas, otra alternativa es mediante el uso de una tabla que ilustre el corrimiento o la sustitución arbritaria escogida, y cuando se deséara cifrar o descifrar un mensaje, se requerirá consultar la correspondencia definida. En nuestro trabajo usamos un conjunto de pares ordenados, que finalmente contienen la misma información que una tabla.

Igualmente importante, la sustitución de un caracter por otro puede ser muy diverso, nosotros incluimos un método porque así cubriremos los objetivos de nuestro trabajo, pero de manera genérica, independientemente de cómo se realicen las sustituciones, nosotros consideraremos que éstas serán *caracter por caracter*. Por lo anterior: aquellos criptosistemas que tengan esta filosofía llevarán el nombre de *cifrados de sustitución monoalfabética*.

Las funciones de cifrado y descifrado que hemos mostrado son un ejemplo concreto de los cifrados de sustitución monoalfabética. por supuesto existen un sinnúmero, y por ello, escogeremos un camino en donde cada función de cifrado se complique hasta llegar de manera natural a nuestro objetivo: *Sustitución polialfabética* que consiste en aplicar diversas sustituciones a un mismo caracter.

2.2.3 Teorema de Euler

Ya que hemos observado que la suma módulo n de k unidades da por resultado un corrimiento, el siguiente paso es el de estudiar los efectos cuando la k se multiplicada módulo n con cada uno de los elementos del mensaje.

Iniciemos nuestra explicación mediante un ejemplo, supongamos que nuestro mensaje a transmitir es $M = \{0, 1, 2, 3\}$, que equivale al conjunto SCR módulo 4, y consideremos dos casos: $k = 2$ y $k = 3$, donde los textos cifrados correspondientes serán:

Caso 1:
$$0 = (0)(\mathbf{2}) \bmod 4 \quad 2 = (1)(\mathbf{2}) \bmod 4$$
$$0 = (2)(\mathbf{2}) \bmod 4 \quad 2 = (3)(\mathbf{2}) \bmod 4$$

Caso 2:
$$0 = (0)(\mathbf{3}) \bmod 4 \quad 3 = (1)(\mathbf{3}) \bmod 4$$
$$2 = (2)(\mathbf{3}) \bmod 4 \quad 1 = (3)(\mathbf{3}) \bmod 4$$

En el primer caso, $C = \{0, 2, 0, 2\}$, en donde solamente aparecen dos elementos del SCR; es decir, que a dos textos en claro distintos les ha correspondido el mismo texto en cifrado, por lo que no se cumple las propiedades de las funciones de cifrado vistas en la sección 1.3

Mientras que para el caso 2, $C = \{0, 3, 2, 1\}$, y cada uno de los elementos cifrados han sufrido un corrimiento distinto al que estudiamos en la sección anterior, por lo tanto veamos lo que se puede aprovechar de esta observación para definir una función de cifrado que proponga una sustitución diferente del corrimiento.

Como los textos, tanto cifrados como en claro, se obtienen a partir de la alteración de cada uno de sus caracteres, los cuales corresponden a un entero del conjunto SCR módulo m. Por lo tanto vamos a definir como el dominio de nuestras funciones al conjunto SCR módulo m, el cual notaremos como $\mathbb{Z}_m$.

Si el emisor y el receptor acuerdan como dominio y contradominio al conjunto $\mathbb{Z}_4$ y acuerdan una llave $k = 2$, como vemos en el Caso 1, al recibir el entero 0, no se sabrá de que entero es producto, ya que existirán dos opciones.

Supongamos ahora que acuerdan una llave $k = 3$; como ilustramos en el Caso 2, no se presenta algún problema, sin embargo ahora la dificultad radica en encontrar la función de descifrado. Por ejemplo, al recibir $c_i = 2$, se sabe que: $(m_i)(3) \equiv 2 \bmod 4$, para algún m_i, por lo tanto, la meta es encontrar el m_i adecuado. Una forma de hacerlo es mediante ensayo y error.

Nosotros sabemos que uno de los principales objetivos de las matemáticas es la de simplificar los problemas, por lo que mostraremos otro camino mucho más eficiente y general. Para comenzar vamos a introducir un resultado que nos servirá a lo largo de la sección:

$$\begin{aligned} a &\equiv b \bmod m \\ &\Rightarrow ac \equiv bc \bmod m, \forall c \in \mathbb{Z} \end{aligned}$$

Para convencerse del resultado únicamente apliquemos nuestra definición de congruencia:

$$\begin{aligned} a &\equiv b \bmod m \\ &\Rightarrow m|(a-b) \\ &\Rightarrow a - b = mq,\ q \in \mathbb{Z} \\ &\Rightarrow ac - bc = mcq, \text{ donde } q_1 = cq \in \mathbb{Z} \\ &\Rightarrow ac - bc = mq_1 \\ &\Rightarrow m|(ac-bc) \\ &\Rightarrow ac \equiv bc \bmod m \end{aligned}$$

También es importante hablar del inverso multiplicativo a^{-1} de un número $a \neq 0$, cuya propiedad consisten en:

$$(a)(a^{-1}) = 1$$

En el caso de las congruencias, no tiene sentido preguntarse sobre la igualdad, es decir, el inverso multiplicativo de un entero $a \neq 0$ módulo m, es aquel entero a^{-1} con la siguiente propiedad:

$$(a)(a^{-1}) \equiv 1 \bmod m.$$

Gracias a que multiplicar por un entero no modifica la congruencia origial y utilizando la definición de inverso multiplicativo, el problema del comienzo se replantea de la siguiente forma:

$$\begin{aligned} (m_i)(3) &\equiv 2 \bmod 4 \\ &\Rightarrow (3^{-1})(m_i)(3) \equiv (3^{-1})(2) \bmod 4 \\ &\Rightarrow m_i \equiv (3^{-1})(2) \bmod 4 \end{aligned}$$

El problema entonces se simplifica en encontrar $a^{-1} = 3^{-1}$, en un inicio ya propusimos una forma de encontrarlo: mediante el ensayo y error encontramos el entero a^{-1} para lo cual $(a^{-1})(3) \equiv 1 \bmod 4$.

La meta en la presente sección es la de ilustrar la solución para el caso general, es decir, replantearemos el problema anterior para definir cuándo existe el inverso de un número módulo m y la forma de determinarlo.

Para comenzar requerimos introducir el *Teorema de cancelación*, para nuestro objetivo, sin pérdida de generalidad vamos a concentrarnos en un caso particular del teorema, cuando $d = 1$:

Si $ab \equiv ac \bmod m$ y $(a, m) = 1$ entonces $b \equiv c \bmod m$.

La afirmación se confirma al observar que: $m|a(b-c)$ y como $(m,a) = 1$ implica que no existe algún primo en común entre a y m, es decir, $m \not| a$, sin embargo m divide al producto $a(b-c)$ y por lo tanto imprescindiblemente $m|(a-b)$.

Por ejemplo, nosotros sabemos que $(2,9) = 1$, lo que implica poder cancelar el 2 cuando $m = 9$, mientras que $(2,14) \neq 1$, al cancelar el 2 usando $m = 14$ los enteros involucrados ya *no* serán congruentes, para lo cual utilizamos el símbolo $\not\equiv$:

$$\begin{aligned} 38 &\equiv 2 \bmod 9 \\ &\Rightarrow (19)(2) \equiv 2 \bmod 9 \\ &\Rightarrow 19 \equiv 1 \bmod 9 \\ 40 &\equiv 26 \bmod 14 \\ &\Rightarrow (2^3)(5) \equiv (13)(2) \bmod 14 \\ &\Rightarrow (4)(5) \not\equiv 13 \bmod 14 \end{aligned}$$

Ahora vamos a definir un nuevo conjunto, que llamaremos *Sistema Reducido de Residuos* módulo m (SRR) conformado por aquellos $r_i \in \mathbb{Z}_m$ tales que $(r_i, m) = 1$, es decir, cada elemento del SRR es primo relativo con el módulo utilizado. Y además vamos a considerar que el conjunto SRR contiene $\varphi(m)$ elementos.

Habíamos mostrado en la sección 2.2.1 que si dos enteros, por decir a y b, son primos relativos respecto a otro, que llamos c, entonces el producto ac sigue siendo primo relativo respecto b.

El argumento anterior es fundamental para comprender que cada elemento $X_i \in SRR$ que es primo relativo con el módulo m, será de igual forma primo relativo con m aunque lo multipliquemos por algún entero a que también es primo relativo al módulo, es decir el conjunto $(a)(X_i)$, contiene los elementos que son primos relativos con m.

Para mostrar que el conjunto obtenido también contien elementos que son primos relativos al módulo, utilicemos el Teorma de cancelación: si consideramos que $aX_i \equiv aX_j \bmod m$, como $(a, m) = 1$ sabemos que es factible cancelar el entero a, con ello obtenemos $X_i \equiv X_j \bmod m$, lo cual es válido solamente cuando $i = j$.

En otras palabras, estamos diciendo que si cada elemento del conjunto SRR es multiplicado por algún entero a, que también es primo relativo al módulo m, entonces dichos elementos también serán primos relativos al módulo.

Con el resultado anterior, para acercarnos a nuestra meta final, a partir de la hipótesis de que $(a, m) = 1$ y considerando cada $X_i \in SRR$, vemos que cada X_i será congruente con algún $(a)(X_j)$, por lo tanto consideremos el producto de todos los elementos de cada uno de los conjuntos.

$$(a)(X_1)(a)(X_2)\ldots(a)(X_{\varphi(m)}) \equiv (X_1)(X_2)\ldots(X_{\varphi(m)}) \bmod m$$

Cada elemento del conjunto SRR es primo relativo a m, porque para que X_i pertenezca al conjunto SRR $(X_i, m) = 1$, por lo tanto podemos cancelar cada X_i de la expresión anterior, es decir:

$$(a)(a)\ldots(a) \equiv 1 \bmod m,$$

Como solamente existen $\varphi(m)$ elementos en SRR concluimos:

$$a^{\varphi(m)} \equiv 1 \bmod m.$$

El resultado que hemos mostrado lleva el nombre de *Teorema de Euler*, gracial a él tenemos un mecanismo más eficiente para encontrar los inversos multiplicativos módulo m. Utilizando la siguiente ecuación veremos concretamente la razón de la explicación mostrada:

$$\begin{aligned}
(a, m) &= 1 \\
ax &\equiv 1 \bmod m \\
a^{\varphi(m)} &\equiv 1 \bmod m \\
&\Rightarrow ax \equiv a^{\varphi(m)} \bmod m \\
&\Rightarrow x \equiv a^{\varphi(m)-1} \bmod m
\end{aligned}$$

Como a y m son primos relativos no hay problemas al dividir entre $a \neq 0$ y finalmente conocer el inverso multiplicativo, indicado en la ecuación anterior por el entero x. Ahora entendemos porque en el Caso 1, del ejemplo mostrado al inicio de esta sección, no existe inverso multiplicativo para el entero 2, módulo 4. Es decir, como $(4,2) \neq 1$ entonces no existe el inverso multiplicativo para el entero 2 módulo 4. Mientras que en el Caso 2, dado que $(3,4) = 1$, entonces *si* existe el inverso multiplicativo para 3 módulo 4, y para encontrarlo tendremos que seguir los siguientes pasos:

$$\begin{aligned}(3)(3^{-1}) &\equiv 1 \bmod 4 \\ &\Rightarrow (3)3^{\varphi(4)-1} \equiv 1 \bmod 4 \\ &\Rightarrow (3)(3) \equiv 1 \bmod 4\end{aligned}$$

Es decir, el inverso de 3 módulo 3 es el entero 3. Aunque los enteros con los que ilustramos las operaciones son muy pequeños, contar los elementos de SRR fué simple, es decir, para conocer $\varphi(m)$ hemos contado todos los divisores primos relativos con $m = 4$, sin embargo si imaginamos que $m = 298384$, entonces nos enfrentaremos con una labor que necesitará bastante tiempo.

Otra forma de determinar $\varphi(m)$ consiste en aplicar la siguiente fórmula que se le conoce con el nombre de *Función de Phi de Euler*:

$$\varphi(m) = \prod_{i=1}^{t} P_i^{e_i-1}(P_i - 1)$$

Es decir, para conocer el valor de $\varphi(m)$ es importante conocer la factorización prima de m, y con ello conoceremos los primos con sus correspondientes exponentes. Lo que la fórmula $\varphi(m)$ indica, corresponde a las siguientes operaciones: Calculamos el producto $(P_1^{e_1-1})(P_1^{e_1} - 1)$ y realizamos el mismo procedimiento con los t primos involucrados en la factorización de m. Cada uno de los productos obtenidos ahora los multiplicamos entre si.

Lo que en palabras parece muy complicado de entender, en la notación matemática se reduce; ya que si conocemos que:

$$m = (P_1^{e_1})(P_2^{e_2}) \ldots (P_{(t-1)}^{e_{(t-1)}})(P_t^{e_t})$$

Lo que hay que hacer es calcular el producto:

$$\varphi(m) = (P_1^{e_1-1})(P_1^{e_1} - 1)(P_1^{e_2-1})(P_2^{e_2} - 1) \ldots (P_1^{e_t-1})(P_t^{e_t} - 1)$$

Pero para indicar el producto de los t primos, nos auxiliamos del símbolo $\prod_{i=1}^{t}$, que equivale al producto de t elementos, y para identifcar cada uno de los elementos, utilizamos el índice i. Por ejemplo:

$$\begin{aligned}
m &= P_1^1 \\
&= 5 \\
\varphi(5) &= (P_1^{1-1})(P_1 - 1) \\
&= 5^{1-1}(5-1) \\
&= 4 \\
m &= P_1^3 \\
&= 2^3 \\
\varphi(8) &= (P_1^{3-1})(P_1 - 1) \\
&= 2^{3-1}(2-1) \\
&= 2^2 \\
m &= (P_1^2)(P_2^1) \\
&= (2^2)(3) \\
\varphi(12) &= (P_1^{2-1})(P_1 - 1)(P_2^{1-1})(P_2 - 1) \\
&= 2^{2-1}(2-1)3^{1-1}(3-1) \\
&= (2)(2)
\end{aligned}$$

Función de cifrado mediante producto módulo m

Con las bases que hemos expuesto, propongamos una función de cifrado. que consista en realizar el producto de un entero con cada uno de las componentes del mensaje. Es decir, cada c_i del mensaje en claro estará definido como:

$$\begin{aligned} c_i &= E_k(m_i) \\ &= (m_i)(k) \bmod m \end{aligned}$$

Y para obtener cada m_i del mensaje en claro, la función de cifrado para cada componente del texto cifrado estará definido del siguiente modo:

$$\begin{aligned} m_i &= D_k(c_i) \\ &= (c_i)(k^{-1}) \bmod m \end{aligned}$$

Para que la expresión anterior sea una propuesta corresta, es importante considerar que:

$$(k, m) = 1,$$

Si queremos conocer el tamaño del espacio de llaves, es importante conocer las llaves k tales que su valor es primo relativo al módulo utilizado, por ejemplo, si el módulo es 26, el número de llaves corresponde al valor de $\varphi(26)$, es decir:

$$\begin{aligned} \varphi(26) &= 13^{1-1}(13-1)2^{1-1}(2-1) \\ &= 12 \end{aligned}$$

Al igual que en el cifrado, mediante la suma módulo m. con el producto obtendremos una nueva sustitución del alfabeto, y por ello, éste cifrado también pertenece a aquellos que nombramos como *sustitución monoalfabética.*

Para conocer una sustitución en particular, consideremos que cada una de las letras está representado por un entero, al igual que lo hicimos en la sección 2.2.2, y escogamos $k = 3$, dado que $(3, 26) = 1$ significa que la llave escogida tendrá su inverso multiplicativo. Las letras del alfabeto en claro, las escribimos en la primera y tercera línea, mientras que las letras cifradas se encuentran en la segunda y cuarta linea del siguiente arreglo:

0	1	2	3	4	5	6	7	8	9	10	11	12
0	3	6	9	12	20	18	21	24	1	4	7	10
13	14	15	16	17	18	19	20	21	22	23	24	25
13	16	19	22	25	2	5	8	11	14	17	20	23

Lo que hemos obtenido ha sido una **nueva** sustitución de los caracteres, ya que ahora, A=A, B=C, C=G, etc. Es importante resaltar que la nueva sustitución no ha sido resultado de un simple corrimiento. Nuestro siguiente interés es el de relacionar ambos resultados, es decir, proponer una función de cifrado que involucre tanto la suma como el producto módulo m.

Función de cifrado mediante producto y suma módulo m

Mediante un corrimiento, obtenido mediante la suma módulo m, en la sección anterior indicamos que existían 26 llaves posibles, y si se trataba de una función aplicando el producto módulo m, entonces eran 12 llaves (considerando un módulo igual a 26).

En la función de cifrado obtenida por el producto módulo m, observamos que el número de llaves considerando $m = 26$ es menor en comparación con la simple suma módulo m, por lo tanto, nuestro objetivo ahora es el de proponer una función que interrelacione ambos resultados, es decir:

$$\begin{aligned} c_i &= E_{k_1,k_2}(m_i) \\ &= (m_i)(k_1) + k_2 \bmod m \\ m_i &= D_{k_1,k_2}(c_i) \\ &= ((c_i) - k_2)(k_1)^{-1} \bmod m \end{aligned}$$

Ahora el número de elementos de $\mathcal{K}$ es de $(12)(26) = 312$, lo que representa un incremento significativo. Para ilustrar su funcionamiento cifremos un mensaje M con $k_2 = 3$ y $k_1 = 5$ (Observemos que $(k_1, m) = 1$).

$$\begin{aligned} C &= \{E_{5,3}(\text{ ESTO SE COMPLICA })\} \\ &= \{E_{5,3}(m_1), E_{5,3}(m_2), \ldots, E_{5,3}(m_{13}), E_{5,3}(m_{14})\} \\ &= \{(m_1)(5) + 3 \bmod 26, \ldots, (m_{14})(5) + 3 \bmod 26\} \\ &= \{(4)(5) + 3 \bmod 26, \ldots, (0)(5) + 3 \bmod 26\} \\ &= \{23, 15, \ldots, 13, 3\} \\ &= \{c_1, c_2, \ldots, c_{13}, c_{14}\} \\ &= \{\text{ XPUVPXNVLAGRND }\} \end{aligned}$$

Para determinar el texto en claro, necesitaremos del inverso multiplicativo de 5 módulo 26, como escogimos un entero primo relativo a 26, la tarea consiste en determinar $(5)(5^{\varphi(26)-1}) \equiv 1 \bmod 26$:

$$\begin{aligned} 5^{-1} &\equiv 5^{\varphi(26)-1} \bmod 26 \\ &\equiv 5^{12-1} \bmod 26 \\ &\equiv 5^{11} \bmod 26 \\ &= 21 \\ &\Rightarrow (5)(21) \equiv 1 \bmod 26 \end{aligned}$$

Por lo tanto la función de descifrado para el ejemplo, se encuentra dada del siguiente modo:

$$\begin{aligned} M &= \{D5, 3(C)\} \\ &= \{D_{5,3}(c_1), D_{5,3}(c_2), \ldots, D_{5,3}(c_{13}), D_{5,3}(c_{14})\} \\ &= \{(c_1 - 3)21 \bmod 26, (c_2 - 3)21 \bmod 26, \ldots, (c_{14} - 3)21 \bmod 26\} \\ &= \{(23 - 3)21 \bmod 26, \ldots, (3 - 3)21 \bmod 26\} \\ &= \{4, 18, 19, \ldots, 8, 2, 0\} \\ &= \{m_1, m_2, m_3, \ldots, m_{12}, m_{13}, m_{14}\} \\ &= \{ \text{ ESTOSECOMPLICA } \} \end{aligned}$$

Para concluir con los fundamentos matemáticos utilizados en este trabajo, solamente nos resta hablar de algunos conceptos del Álgebra.

2.3 Álgebra

En las secciones anteriores mostramos algunas aplicaciones empleando aritmética modular; sin embargo, para conocer su verdadero potencial es necesario reflexionar otros aspectos, los cuales abordaremos en ésta sección.

En las matemáticas no interesa resolver únicamente problemas particulares, sino que se buscan encontrar generalidades más ambiciosas cuya aplicación sea poderosa, es por ello que preferimos hacer una pausa y mostrar la estructura matemática involucrada en las congruencias.

Por estructura nos referimos a características muy concretas que comparten los entes matemáticos, con ello cuando sea necesario utilizarlos sabremos el verdadero alcance y limitaciones que se tienen con su uso.

Por ejemplo, si conocemos la estructura que tiene un automóvil, cuando se utilice uno se sabrá que no vuela y no habrá confusiones con un burro o un camello, ni existirá confusión con las diversas marcas de carros existentes.

Cuando se desconoce la estructura de un ente que estamos utilizando, nos enfrentamos a una variedad de casos que no ayudan a reconocer nuevos horizontes. Regresando a nuestro ejemplo, de no conocer la estructura de un automóvil, invertiremos tiempo y esfuerzo en conocer un Volkswagen y cuando veamos un Fórmula 1 nos cuestionaremos si ésta máquina puede planear por los cielos.

En nuestro trabajo, cuando realizamos operaciones aritméticas en $\mathbb{Z}_m$ en ninguna parte mostramos la estructura utilizada, a pesar de ser un concepto nuevo veremos que inclusive en $\mathbb{Z}$ también se encuentran problemas conceptuales a raíz de la poca reflexión en su estudio.

Para ilustrar nuestro argumento, usemos elementos muy bien manejados por todos nosotros: "1 más 1 es igual a 2". Nadie nos explicó la razón de dicha verdad.

Si la operación anterior no involucrara números, poco se podría concluir, dado que nuestro conocimiento adquirido es producto de la mecanización; por ejemplo, si viajáramos a un mundo desconocido y escuchamos que alguien dice "grl m grl igual a ki", nuestra expresión de desconcierto saltaría a la vista al ignorar si están sumando, si acaso es otra representación de los números o si es una operación nueva. Lo único que se concluye es que los habitantes tienen una comunicación en donde si alguien dice *grl m grl* se contesta *ki*.

Si en nuestra educación ignoráramos los aspectos no razonados y mejor aprendiérmos la estructura utilizada, tendríamos la capacidad de explicar un poco más sobre lo que sucede en este planeta extraño en el que nos encontramos y mejor que eso: el interés por las matemáticas sería mayor.

Otro de los problemas asociados con nuestro conocimiento matemático, es que suponemos que no existen más operadores de los que el profesor de primaria nos ilustró; seguramente si un alumno propusiera el operador "!", de tal forma que si tomamos un par de números a y b $a \; ! \; b$ es igual al menor entre los dos, lo ignorarían.

No debe de ser extraño el operador !, ya que lo usamos a diario: cuando vamos al supermercardo y vemos dos barras de granola, una que cuesta 40 y otra 9 pesos, si queremos llevarnos la más barata debemos escoger la del menor precio, es decir, mentalmente calculamos $40 \; ! \; 9 = 9$.

Si no conocemos el operador !, entonces realizaremos una elección de compra incorrecta. Así como este operador, en nuestra vida cotidiana usamos otros sin saberlo; del mismo modo, intuitivamente sabemos los factores que se involucran para un operador dado; por ejemplo, no hay duda que es un error decir: "1 más perro".

Para continuar provechosamente, es necesario vislumbrar el significado de "un operador binario cualquiera (*)". Por definición [FRA87] un operador binario *, empleado en un conjunto de objetos, es una función que asigna a cada par ordenado otro elemento del conjunto.

Es importate considerar un par ordenado porque sabemos que $(a, b) \neq (b, a)$ si $a \neq b$, Quiere decir que no necesariamente los operadores deben de ser *conmutativos*: si definimos el operador "@" como $a \ @ \ b = a$, el resultado difiere si escribimos $b \ @ \ a$.

El operador @ se encuentra vinculado con el orden, operador que usamos cuando nos vestimos y como bien sabemos no es lo mismo ponerse los calcetines y luego los zapatos, que los zapatos y luego los calcetines.

Además el operador binario tiene por resultado un elemento de la misma naturaleza de los que se están utilizando: si los objetos son los número enteros sabemos que "1+1" no es 'casa', porque *casa* no pertenece al conjunto $\mathbb{Z}$, al igual que no tiene sentido sumar "peras + manzanas".

El objetivo del Álgebra entonces, es el de estudiar conjuntos de objetos abstractos juntos con una operación binaria, que definirá una estructura en particular. El resultado se refleja en obtener la comprensión de lo que hasta ahora se manejaban con ejemplos.

No es importante definir cuáles son los objetos abstractos que serán utilizados por el operador *, dado que en el Álgebra no hay una tarea en mostrar los múltiples objetos a utilizar, es por ello que vamos a realizar la abstracción considerando cualesquiera.

Con el preámbulo anterior, discutiremos los aspectos básicos de la estructura llamada *grupo*, con el que cerraremos los fundamentos matemáticos necesarios y además mencionaremos un criptosistema de llave pública.

2.3.1 Grupos

Existe un inmenso material relacionado con el estudio de los *grupos* en el Álgebra. Para comenzar es importante conocer la definición de lo que estudiaremos [FRA87]:

Un grupo $< G, * >$ es un conjunto G no vacío junto con una operación binaria * en G, tal que satisface los siguientes puntos:

1. La operación binaria * es asociativa
2. Existe un elemento e en G tal que $e * x = x * e = x$, $\forall x \in G$
3. Para cada a en G existe un elemento a^{-1} con la propiedad de que $a^{-1} * a = a * a^{-1} = e$

El elemento e lleva el nombre de **elemento identidad** para * en G, mientras que a a^{-1} se le denomina **inverso de a respecto** a *.

Un ejemplo muy simple es el conjunto $\mathbb{Z}$ con el operador *suma*. Para afirmar que el conjunto de los enteros junto con la suma es un grupo debemos verificar si se cumple la definición de grupo, es decir:

1. $(a + b) + c = a + (b + c)$ con $a, b, c \in \mathbb{Z}$
2. $a + 0 = 0 + a = a$. La identidad es el 0.
3. $a + (-a) = 0$. El inverso aditivo de cualquier entero a es $-a$

Sin embargo, si consideramos otra operación binaria no necesariamente se mantiene la estructura de grupo; por ejemplo, si ahora el operador binario es la división observamos que:

1. La operación binaria no es cerrada: Por ejemplo si tomamos al entero 5 y 2, 5/2 no es un número entero.

Por lo tanto, no tiene sentido preguntar si un conjunto por si solo es un grupo, sino que deberemos considerar el operador binario que se está utilizando.

Los grupos que nos interesaran en nuestro trabajo, son aquellos en donde intervienen conjuntos finito. Para mostrar algunos ejemplos, será importante comenzar nuestra reflexión construyendo el grupo finito más pequeño; para hacerlo nos remitiremos a nuestra definición en donde se nos indica la existencia de un elemento llamado identidad e:

Ahora verifiquemos si con el elemento e y un operador * es posible contruir un grupo:

1. $e * (e * e) = (e * e) * e$: Se cumple la asociatividad
2. $e * e = e$: Existe la identidad, que es precisamente e.
3. $e * e = e$: Existe el inverso respecto a * para e, el cual es precisamente el mismo elemento e.

Si deseamos construir un conjunto más grande que también utilice la operación binaria *, agregemos un elemento más al conjunto anterior, para tener:

$$\{e, a\}$$

Para construir una estructura de grupo deberemos definir la operación binaria del siguiente modo:

$$\begin{aligned} e * e &= e \\ e * a &= a \\ a * e &= a \\ a * a &= e \end{aligned}$$

Con lo anterior hemos ilustrado un ejemplo de un grupo finito. Y ésto nos lleva a introducir un nuevo término; cuando empleamos conjuntos finitos *el orden* $|G|$ de G es el número de elementos en G [CAR90]. Por ejemplo, en el caso anterior el *orden del grupo* es de 2.

Para ilustrar la importancia del estudio de los grupos, supongamos que llega a nuestras manos la siguiente tabla:

$$\begin{aligned} 0+0 &= 0 \\ 0+1 &= 1 \\ 1+0 &= 1 \\ 1+1 &= 0 \end{aligned}$$

Que es similar a la tabla correspondiente al grupo de orden 2 que incluimos, con la diferencia de que $0 = e$ y $1 = a$. A excepción del nombre de los elementos de cada conjunto, practicamente podemos afirmar que son el mismo grupo de orden 2. En una situación similar afirmaremos que dichos grupos son "estructuralmente iguales",

El grupo que nos interesa, porque lo hemos trabajado en las secciones anteriores, es $\mathbb{Z}_m$ con el operador suma, veamos que dicho conjunto con el operador suma tiene una estructura de grupo:

1. $(a+b) \bmod m + c \bmod m = a \bmod m + (b+c) \bmod m$, con $a, b, c \in \mathbb{Z}$: Se cumple la asociatividad.

2. $a + 0 \bmod m = a \bmod m$: el elemento 0 es la identidad.

3. $a + (-a) \bmod m = 0$: El inverso aditivo del a es $-a$.

Considerando el grupo anterior, observemos el resultado al sumar iteradamente el elemento 1:

$$\begin{aligned} 1 &= 1 \bmod m \\ 2 &= 1+1 \bmod m \\ 3 &= 1+1+1 \bmod m \\ 4 &= 1+1+1+1 \bmod m \\ &\vdots \quad \vdots \quad \vdots \\ m-1 &= \overbrace{1+1+\ldots+1+1}^{(m-1)veces} \bmod m \\ 0 &= \overbrace{1+1+\ldots+1+1}^{(m)veces} \bmod m \\ &\equiv m \bmod m \end{aligned}$$

Como observamos, los efectos obtenidos al sumar iteradamente el 1 m veces es el de obtener el conjunto $\mathbb{Z}_m$. Cuando ésto sucede, en el caso particular del ejemplo anterior, diremos que el *1 es un generador*.

Sin embargo no siempre se tiene al elemento 1, es posible encontrar otro, por ejemplo, si emplemos $\mathbb{Z}_4$ y veamos que el entero 4 es un generador para el conjunto $\mathbb{Z}_4$ y el operador suma:

$$\begin{aligned} 3 &= 3 \bmod 4 \\ 2 &= 3+3 \bmod 4 \\ 1 &= 3+3+3 \bmod 4 \\ 0 &= 3+3+3+3 \bmod 4 \end{aligned}$$

Si pensamos ahora en la multipicación, como indicamos en la sección 2.2.3, no todo entero $a \in \mathbb{Z}_n$ tiene un inverso multiplicativo. Sin embargo, mostramos que el inverso multiplicativo existe si y solamente si, el entero dado es primo relativo respecto al módulo.

Para construir un grupo con el conjunto $\mathbb{Z}_n$ y utilizar el producto como operador binario, es posible hacerlo si consideramos que el módulo es un número primo p. Dado que un número primo es divido por el 1 y el mismo primo, entonces los enteros que se encuentran entre 1 y $p-1$ són primos relativos con p.

Para verificar $\mathbb{Z}_p$, junto con la multiplicación es un grupo, tenemos que comprobar lo siguiente:

1. $(a * b) \bmod p * c \bmod p = a \bmod p * (b * c) \bmod p$, con $a, b, c \in \mathbb{Z}$: Se cumple la asociatividad.

2. $a * 1 \bmod p = 1 * a \bmod p$: el elemento 1 es la identidad.

3. $a * (a^{-1}) \bmod p = 1$: El inverso multiplicativo de a es a^{-1}: El inverso multiplicativo existe, dado que todo entero $a \neq p$ es primo relativo con p.

Por lo tanto, al igual que hemos encontrado para la suma módulo n un generador, también lo podemos hacer para el producto módulo p; por ejemplo, consideremos el siguiente conjunto:

$$Z_{31}^{*} = \{1, 2, \ldots, 30\}$$

Utilizando el producto módulo $p = 31$, después de un poco de trabajo encontramos que $\alpha = 13$ cumple con la cualidad de ser un generador para éste conjunto empleando el producto módulo 31.

Para verificar que $\alpha = 13$ es un generador, al igual que sumamos interadamente, ahora vamos a multiplicar iteradamente este elemento, de donde obtenemos el siguiente resultado (Tabla 2.3.1):

Multiplicación iterada de 13 módulo 31

Tabla 2.3.1

$13 = 13^{01} \bmod 31$	$14 = 13^{02} \bmod 31$	$27 = 13^{03} \bmod 31$	$10 = 13^{04} \bmod 31$
$06 = 13^{05} \bmod 31$	$16 = 13^{06} \bmod 31$	$22 = 13^{07} \bmod 31$	$07 = 13^{08} \bmod 31$
$29 = 13^{09} \bmod 31$	$05 = 13^{10} \bmod 31$	$03 = 13^{11} \bmod 31$	$08 = 13^{12} \bmod 31$
$11 = 13^{13} \bmod 31$	$19 = 13^{14} \bmod 31$	$30 = 13^{15} \bmod 31$	$18 = 13^{16} \bmod 31$
$17 = 13^{17} \bmod 31$	$04 = 13^{18} \bmod 31$	$21 = 13^{19} \bmod 31$	$25 = 13^{20} \bmod 31$
$15 = 13^{21} \bmod 31$	$09 = 13^{22} \bmod 31$	$24 = 13^{23} \bmod 31$	$02 = 13^{24} \bmod 31$
$26 = 13^{25} \bmod 31$	$28 = 13^{26} \bmod 31$	$23 = 13^{27} \bmod 31$	$20 = 13^{28} \bmod 31$
$12 = 13^{29} \bmod 31$	$01 = 13^{30} \bmod 31$		

2.4 Criptosistema de llave pública: ElGamal

Para aterrizar todos los conceptos que hemos estudiado ahora vamos a exponer un criptosistema de llave pública que lleva el nombre de: *ElGamal*

Como lo indicamos en la sección 1.6, en los criptosistemas de llave pública es indispensable definir, en primer término, el procedimiento a seguir para generar el par de llaves necesarias. Similarmente como en las secciones anteriores, nuetros interlocutores los identificaremos como A y B.

1. A y B conocen un primo p y un generador $\alpha \in Z_p^* = \{1, 2, \ldots, (p-1)\}$
2. Cada uno escoge su llave privada $d_A, d_B \in Z_p^*$
3. Calculan su llave pública: $e_A \equiv \alpha^{d_A} \bmod p$ y $e_B \equiv \alpha^{d_B} \bmod p$

Tanto el generador, el número primo y el valor e_A y e_B son valores conocidos, es decir, corresponden a la llave pública. El éxito del sistema se encuentra en no poder encontrar el exponente x de la congruencia $e_A \equiv \alpha^x \bmod p$, dicho planteamiento lleva el nombre de *problema del logaritmo discreto.*

Dicho en otros términos, hasta el momento ha sido computacionalmente imposible encontrar el exponente de α sabiendo la llave pública. Claro, ésto es válido cuando p es grande; por ejemplo si consideramos $p = 31$, en la tabla 2.3.1 hemos incluido todos los diversos valores del generador, es decir, conocemos el valor de x para cualquier $e_A \equiv \alpha^x \bmod p$, sin embargo cuando p es muy grande entonces la situación es diferente. Por supuesto que para dar un argumento matemático de la sentencia anterior, es importante remitirnos a la complejidad computacional, tema que no corresponde a nuestro trabajo.

Por otra parte, para facilitar la comprensión del criptosistema ElGamal, todos los cálculos considerarán el primo $p = 31$ y $\alpha = 13$, porque en la tabla 2.3.1 ya hemos efectuado todos los calculos necesarios.

Para entrar en materia, indiquemos los pasos necesarios en el criptosistema ElGamal:

1. A escoge $d_A = 17$ y B $d_B = 22$
2. Cada quien determina su llave pública:
   - $e_A \equiv \alpha^{d_A} \bmod 31$
     $\equiv 13^{17} \bmod 31$
     $= 17 \bmod 31$
   - $e_B \equiv \alpha^{d_B} \bmod 31$
     $\equiv 13^{22} \bmod 31$
     $= 9 \bmod 31$
3. Cada quien da a conocer su llave pública:
   - A conoce $e_B = 9$
   - B conoce $e_A = 17$

Si A desea mandar un mensaje a B, entonces deberá seguir los siguientes pasos:

1. Verificar que $M \in Z_p^*$: Cada mensaje M que se comunique deberá ser un entero que se encuentre dentro del conjunto Z_p^*.
2. Escoger $k \in Z_p^*$
3. Calcular $K \equiv e_B^k \bmod p$
4. Envía $c_1 \equiv \alpha^k \bmod p$ y $c_2 \equiv MK \bmod p$

Siguiendo estos pasos, con los valores que hemos escogido tenemos que A escoge $M = 21$ y con lo fundamentado anteriormente, concluimos que para cifrar debe de cumplirse:

1. $21 \in Z_{31}^*$
2. Escoge $k = 13$
3. Calcula:
$$\begin{aligned} K &\equiv e_B^k \bmod 31 \\ &\equiv 9^{13} \bmod 31 \\ &= 18 \end{aligned}$$
4. Envía :
$$\begin{aligned} c_1 &\equiv \alpha^{13} \bmod 31 & c_2 &\equiv MK \bmod 31 \\ &\equiv 13^{13} \bmod 31 & &\equiv (21)(18) \bmod 31 \\ &= 11 & &= 6 \end{aligned}$$

Como vemos el proceso de cifrado es un poco laborioso, para discutir sobre el descifrado, gracias a lo que hemos presentado con anterioridad, se facilitará considerablemente.

1. B recibe (c_1, c_2)
2. B aplica su llave privada a c_1 para obtener el valor de K
$$\begin{aligned} c_1^{d_B} &\equiv \alpha^{(k)(d_B)} \bmod p \\ &\equiv e_B^k \bmod p \\ &= K \bmod p \end{aligned}$$
3. Determina $K^{-1} \bmod p$
4. $M = c_2 K^{-1} \equiv MKK^{-1} \bmod p$

Con el ejemplo que estamos ilustrando, para descifrar las parejas (c_1, c_2) tenemos que:

1. B recibe $(11, 6)$

2. B aplica su llave privada a 11
$$\begin{aligned} K &\equiv 13^{(k)(17)} \bmod 31 \\ &\equiv 11^{22} \bmod 31 \\ &= 18 \end{aligned}$$

3. $K^{-1} = 19$ (Nota: $(18)(19) \equiv 1 \bmod 31$)

4. $M = 21 \equiv (6)(19) \bmod 31$

Hasta aquí concluimos tanto con nuestro ejemplo de llave púbica, así como con los fundamentos matemáticos; lo que nos resta por hacer es aprovechar lo que hemos desarrollado con el fin de realizar el criptoanálisis a nuestro cifrado de Vigenére

Capítulo 3

Desarrollo

3.1 Cifrado de Vigenére

El autor del cifrado que mostraremos se le atribuye al Francés Blaise de Vigenére, nacido en 1523, quien durante su crecimiento tuvo la suerte de recibir la educación de un noble, lo que le ayudó para desempeñar diversos trabajos en el servicio diplomático, actividad que realizó por un largo tiempo hasta que después tuvo la fortuna de ser enviado a una misión en Roma.

Por su interés en la criptografía, en Roma discutió con diversos expertos en el área los trabajos elaborados por el matemático *J.B Porter*, que giraban alrededor del criptoanálisis de los cifrados de sustitución monoalfabética mediante un análisis estadístico. al igual que otras alternativas que fueron las que inspiraron a Vigenére.

Vigenére regresó a París donde escribió un libro durante sus estudios de criptografía cuya aportación era la *Tabla de Vigenére*, la cual facilitó el uso de un criptosistema que posteriormente llevaría su nombre: *Cifrado de Vigenére*.

Para comprender el criptosistema es necesario remitirnos al mecanismo utilizado a partir de la Figura 3.1. Además, es necesario puntualizar que ahora la llave se conformará de varios elementos. y por ello vamos a emplear como notación la letra K para referirnos a una llave empleada en el cifrado de Vigenére.

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

Figura 3.1: Tabla de Vigenere

El mecanismo es simple y por ello será mejor que lo ilustremos mediante el cifrado de un mensaje corto.

$$K = \{ \text{DILEMA} \}$$
$$M = \{ \text{SER O NO SER HE AHI LA CUESTION} \}.$$

Simplemente para facilitar tanto el cifrado como el descifrado vamos a reescribir la llave y el mensaje en una forma conveniente:

SERONO	SERHEA	HILACU	ESTION
DILEMA	DILEMA	DILEMA	DILEMA

Haciendo uso de la Figura 3.1, ubicamos la primera letra "S" en la primera columna de la Tabla y la letra "D" en el primer renglón: buscamos la intersección y obtenemos "V" que será el texto cifrado asociado para "S" con el elemento de la llave "D".

Tomando el siguiente caracter de la llave y del mensaje, seguimos el mismo proceso hasta que hayamos terminado, y es así como obtenemos el siguiente texto cifrado:

$$C = \{ \text{VMCSZO VMCLQA KQWEOU HAEMAN} \}$$

Para descifrar C es necesario conocer la llave K y del mismo modo, para facilitar la tarea, también nos auxiliaremos de la misma Tabla (Figura 3.1). Ahora indagamos la ubicación de la columna del primer renglón que contenga la letra "D" (primer elemento de la llave). En dicha columna buscamos la letra "V", que es nuestro texto a descifrar y que ha sido cifrado con la letra "D", el texto en claro correspondiente es la letra que se encuentra en la primera columna del renglón en donde se ubica la letra "V". Análogamente el proceso se continúa hasta terminar con el texto cifrado y determinar nuevamente

$$M = \{ \text{SER O NO SER HE AHI LA CUESTION} \}$$

Es interesante observar los distintos valores tomados por cada uno de los caracteres del texto en claro, por ejemplo, la letra "E" ha sido sustituida por las letras: { M, Q, H }. Debido a que un caracter es sustituido por varios, diremos que éste es un *cifrado de sustitución polialfabética.*

En los casos estudiados en la sección 2.2.2 y 2.2.3 habíamos observado que cada caracter tenía *una* sola sustitución asociada y por ello dichos criptosistemas los identificamos con el nombre de *Sustitución monoalfabética*. La variación que hemos hecho ha sido en función del número de sustituación que tiene cada caracter, sin embargo, tanto los cifrados de sustitición monoalfabética como los cifrados de sustitución polialfabética efectuan el cifrado y descifrada letra por letra.

Es de nuestro interés enfocarnos en los criptosistemas de sustitución monoalfabética y polialfabética cuando utilizan la suma módulo m. Dado que en ambos sistemas se necesita una misma llave para cifrar y descifrar, recordemos que la naturaleza de estos criptosistemas los llamamos: Criptosistemas de llave secreta (Ver sección 1.5).

Para lograr la abstración del cifrado de Vigenére; para no emplear la Tabla de Vigenére, del ejemplo con el que iniciamos esta sección, observamos que:

D	I	L	E	M	A
↓	↓	↓	↓	↓	↓
S	E	R	O	N	O
S	E	R	H	E	A
H	I	L	A	C	U
E	S	T	I	O	N

Por lo tanto, las letras del mensaje en claro que se encuentren en la primera columna serán cifradas con la misma letra "D", al observar el texto cifrado C se concluye que todas estas letras han sido corridas 3 posiciones a la derecha. Del mismo modo, aquellas letras cifradas con la letra "I" sufrirán un corrimiento de 8 posiciones a la derecha y así sucesivamente hasta que no existe otra letra en la llave ni texto en claro.

De lo anterior se concluye que el mensaje M ha sufrido 6 corrimientos distintos. En la sección 2.2.2 acordamos que para representar un corrimiento de un entero, que acordamos que corresponde a una letras (ver sección 2.2.2), la suma módulo m nos auxilia.

A lo largo de nuestro trabajo el tamaño de llave lo identificaremos con el nombre de *período*, la notación empleada será mediante la letra D. Por lo tanto, para representar el contenido de la llave K escribiremos:

$$K = \{k_1, k_2, \ldots, k_D\}$$

Donde cada $k_i \in K$ representa un corrimento particular; del ejemplo anterior tenemos:

$$\begin{aligned} K &= \{\text{ D, I, L, E, M, A }\} \\ &= \{3, 8, 11, 4, 12, 0\} \end{aligned}$$

Donde:

$$\begin{array}{lll} k_1 = \text{D} = 3 & k_2 = \text{I} = 8 & k_3 = \text{L} = 11 \\ k_4 = \text{E} = 4 & k_5 = \text{M} = 12 & k_6 = \text{A} = 0 \end{array}$$

Una vez explicado la notación que emplearemos, el proceso de cifrado y descifrado, apoyándonose en la sección 2.2.2, se realiza del siguiente modo:

$$\begin{aligned} C &= \{E_K(M)\} \\ &= \{E_{k_1}(m_1), E_{k_2}(m_2), \ldots, E_{k_D}(m_D), E_{k_1}(m_{D+1}), \ldots, E_{k_D}(m_{2D}), \ldots\} \\ &= \{c_1, c_2, \ldots, c_N\} \\ M &= \{D_K(C)\} \\ &= \{D_{k_1}(c_1), D_{k_2}(c_2), \ldots, D_{k_D}(c_D), D_{k_1}(c_{D+1}), \ldots, D_{k_D}(c_{2D}), \ldots\} \\ &= \{m_1, m_2, \ldots, m_N\} \end{aligned}$$

Una notación compacta para indicar lo anterior, se concentra en el cifrado y descifrado de un caracter en particular con su correspondiente elemento de la llave K, es decir:

$$\begin{aligned} E_{k_i} &= m_j + k_i \bmod m \\ D_{k_i} &= c_j - k_i \bmod m \end{aligned}$$

Los valores que toman tanto m_j como k_i son enteros dentro del conjunto $\mathbb{Z}_m$, aunque es viable utilizar enteros mayores sabemos que éstos se reducen en alguno de los contemplados en SCR.

De la expresiones que hemos incluido, recordemos que m corresponde al número de elementos del alfabeto utilizado. Durante la sección 2.2 siempre trabajamos con un módulo $m = 26$, sin embargo, para ilustrar que es posible utilizar otro módulo, consideremos ahora que $m = 27$, donde incorporamos la letra "ñ" a nuestro alfabeto anterior.

Para motivar el criptoanálisis que cubriremos en la sección siguiente, es interesante recalcar que el número de llaves se ha incrementado considerablemente, ésto lo podemos ilustrar por el hecho de que para la suma módulo m obtenemos un espacio de llaves de 26 elementos, para la suma y producto módulo m consideramos 312 llaves posibles, y con el Cifrado de Vigenére obtenemos 26^D llaves, y si consideramos que el alfabeto tiene 27 elementos, entonces el número de llaves es de 27^D.

Es decir, si tenemos un tamaño de llave $D = 4$ y el módulo $m = 27$, el número de llaves se incrementa enormenente, es decir, el número de elementos de $\mathcal{K}$ es de:

$$\begin{aligned} 27^D &= 26^4 \\ &= 456,976 \end{aligned}$$

Para el criptoanálisis que realizaremos deberemos recordemos (ver sección 2.1.1) que si el número de llaves es tan grande como el número de posibles mensajes, entonces el mensaje es indescifrable. La condición anterior se cumple cuando el tamaño de texto es de N caracteres, y el tamaño de llave es de N caracteres también, es decir, $D = N$.

Para comprender la razón por la cual $D \neq N$. consideremos un ejemplo que ilustre el resultado el resultado de la sección 2.2.1. Consideremos una llave:

$$K = \{ \text{ DILEMA } \}$$

Donde el período de la llave es $D = 6$. Si ahora ciframos un mensaje cuya longitud es $N = 6$, ilustraremos que es imposible encontrar el texto en claro. Por ejemplo, sea M:

$$M = \{ \text{ ESCAPA } \}$$

Cifrando el mensaje anterior mediante el cifrado de Vigenére, tenemos que:

$$\begin{aligned} C &= \{E_K(\text{ ESCAPA })\} \\ &= \{E_{k_1}(\text{E}), E_{k_2}(\text{S}), E_{k_3}(\text{C}), E_{k_4}(\text{A}), E_{k_5}(\text{P})E_{k_D}(\text{A})\} \\ &= \{m_1 + k_1 \bmod 26, \ldots, m_6 + k_6 \bmod 26\} \\ &= \{4 + 3 \bmod 26, 18 + 8 \bmod 26, \ldots, 0 + 0 \bmod 26\} \\ &= \{7, 0, 13, 4, 1, 0\} \\ &= \{c_1, c_2, c_3, c_4, c_5, c_6\} \\ &= \{ \text{ HANEBA } \} \end{aligned}$$

Si intentamos aplicar todas las funciones de cifrado posibles, tendríamos que probar $26^D = 26^6 = 308915776$ llaves factibles. El problema no es el de intentar todas las llaves posibles, sino a la imposibilidad de obtener información alguna. Por ejemplo, si en algun momento utilizamos una llave $K' \in \mathcal{K}$, dada como:

$$\begin{aligned} K' &= \{k'_1, k'_2, k'_3, k'_4, k'_5, k'_D\} \\ &= \{16, 18, 0, 1, 23, 7\} \\ &= \{ \text{ Q S A B X H } \} \end{aligned}$$

Al descifrar C mediante K' tenemos:

$$\begin{aligned} M' &= \{D_{K'}(\text{ HANEBA })\} \\ &= \{D_{k_1'}(\text{H}), D_{k_2'}(\text{A}), D_{k_3'}(\text{N}), D_{k_4'}(\text{E}), D_{k_5'}(\text{B}) D_{k_D'}(\text{A})\} \\ &= \{c_1 - k_1' \bmod 26, \ldots, c_6 - k_6' \bmod 26\} \\ &= \{7 - 16 \bmod 26, 0 - 18 \bmod 26, \ldots, 0 - 7 \bmod 26\} \\ &= \{17, 8, 13, 3, 4, 19\} \\ &= \{m_1', m_2', m_3', m_4', m_5', m_6'\} \\ &= \{\text{ RINDET }\} \end{aligned}$$

Hemos utilizado una K' y observamos que $M' \neq M$ aunque M' tiene sentido. Existen una gran varidedad de posibles mensajes similares a M' y aunque en algún intento, de los 26^D probables, obtengamos $M = \{$ ESCAPA $\}$ no existirán los elementos suficientes para identificar el mensaje correcto.

Como vemos, gracias a las matemáticas hemos llegado a una diversidad de resultados que han facilitado la comprensión del funcionamiento del cifrado de Vigenére. El siguiente punto a cubrir consiste en el criptoanálisis para mostrar que 26^D llaves no es una razón suficiente para afirmar que el criptosistema de Vigenére es seguro.

Por lo tanto recordemos que una hipótesis importante en el criptoanálisis es el de considerar que el período de la llave es extrictamente menor que el número de elementos del texto cifrado.

3.2 Criptoanálisis al Cifrado de Vigenére

Para comprender el criptoanálisis del cifrado de Vigenére, empecemos analizando el caso más simple: El período $D = 1$, que equivale a una sustitución monoalfabética.

En la sección 2.1 exhibimos la distribución de frecuencias relativas (Figura 2.2) de cada una de las letras que intervienen en un mensaje en Español. La frecuencia relativa de una letra dada, corresponde al cociente entre la frecuencia con la que aperece la letra considerada y el número de letras en total.

Por ejemplo, si la letra A aparece 129 veces en un mensaje de 12456 caracteres, la frecuencia relativa es entonces de 0.010. Sin embargo si consideramos a un mensaje más pequeño entonces la frencuencia relativa de la letra puede cambiar.

Por ello en la Figura 2.2 utilizamos un texto de 15,000 letras y además los textos que analizaremos son de longitudes similares, para que los valores de las frecuencias relativas sean similares.

Por último, durante el resto de nuestro trabajo, independientemente de la distancia de Unicidad, definiremos la cantidad de texto cifrado necesario para obtener evaluaciones que permitan encontrar el texto cifrado automáticamente, sin que los factores estadísticos empleados pierdan confianza. Además, vamos a considerar un alfabeto mayor, que será igual al que hemos manejado, agregándole la ñ, es decir intervendrán 27 letras.

Si nosotros conocemos todas las frecuencias relativas de cada una de letra del alfabeto que estemos trabajando, es posible representar cada valor en una misma gráfica, la cual llamaremos *distribución de frecuencias relativas*, que tendrá una forma similar a la Figura 2.2.

Una vez que se conoce el diagrama de distribución de frecuencias relativas de las letras correspondiente a un texto en claro, dicha distribución la llamaremos *patrón de referencia* y en nuestro trabajo equivale a la Figura 2.2. Si obtenemos un texto cifrado, obtenido a partir de una sustitución monoalfabética, el diagrama de distribución de frencuencias relativas de las letras del texto cifrado, deberá ser similar al patrón de referencia, con la diferencia de que cada una de las barras se encontrarán recorridas un cierto número de posiciones [CIP87].

Para explicar nuestro razonamiento hemos escogido aleatoriamente un texto en español de 9,037 caracteres, que cifraremos mediante un corrimiento desconocido. La distribución de frecuencias relativa de cada uno de los caracteres, que los representaremos como enteros (Ver sección 2.2.2), la representamos en la Figura 3.2:

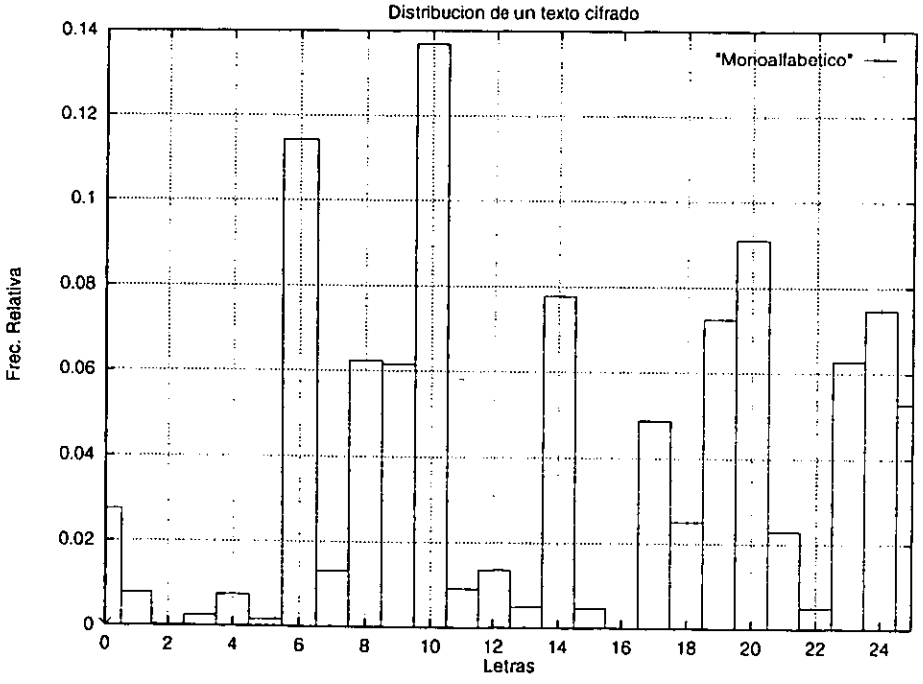


Figura 3.2: Texto cifrado mediante $D = 1$

Al cotejar la Figura 3.2 con la Figura 2.2 resulta que ambas son semejantes, con la diferencia de que la barra marcada con el número 0 ahora se encuentra en el entero 6 en la Figura 3.2. Partiendo de dicha columna además las subsecuentes son similares también: Es decir, si recorremos cada barra de la Figura 3.2 seis unidades a la izquierda, obtenemos la Figura 2.2, por lo tanto la llave de cifrado ha sido $k = 6$.

Aunque sabíamos que es trivial aplicar 26 corrimientos e identificar aquel texto que tuviéra sentido, ahora el trabajo se minimiza considerablemente a raíz de que al graficar el diagrama de frecuencias relativas correspondiente al texto cifrado, como el que hemos mostrado en la Figura 3.2, es necesario contraponerlo con la Figura 2.2. Si el texto cifrado ha sido obtenido a partir de una sustitución monoalfabética entonces el diagrama de frecuencias relativas del texto cifrado estará recorrido un número k de unidades, y precisamente dicho valor corresponde a la llave utilizada [SIN66].

El criptoanálisis que hemos explicado se basa en un *análisis estadístico* y como vimos los resultados se ajustan cuando el período $D = 1$, porque solamente hemos cambiado de nombre a cada una de las letras.

El problema es mucho más complejo cuando una letra se sustituye por diversos valores, como es el caso de la sustitución polialfabética, un análisis estadístico como el realizado no es suficiente.

Por ejemplo, cambiemos el tamaño de llave y consideramos un texto de 13,000 caracteres cifrado a partir de una llave aleatoria de tamaño $D = 19$, como se ve en la Figura 3.3, su distribución de frecuencias relativas no se asemeja a la de una sustitución monoalfabética.

Para justificar el comportamiendo de la Figura 3.3, recordemos que si a cada caracter se le sustituye por diversos caracteres, entonces la frecuencia relativa de éste caracter también se distribuirá entre varios, y por lo tanto obtendremos una distribución talamente diferente.

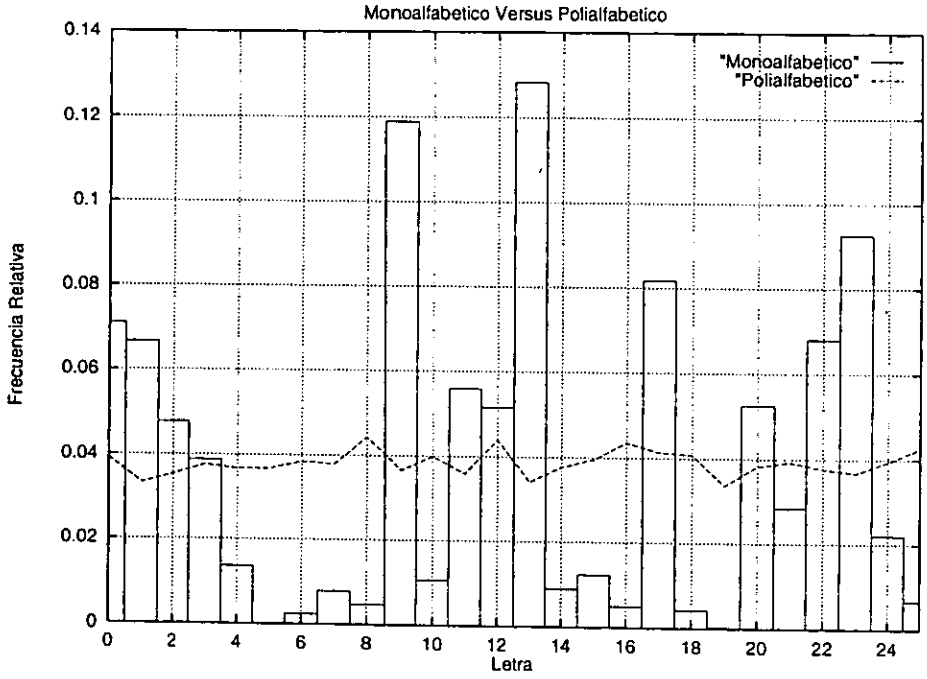


Figura 3.3: Texto Cifrado con $D = 1$ y $D = 19$

Aunque de primera instancia el comportamiento de la distribución de un texto cifrado, producto de una sustitución polialfabética, no se asemeja al de un cifrado de sustitución monoalfabética, la situación cambiaría si el período fuera conocido.

Para proponer un procedimiento, apliquemos el método cuando $D = 1$ a cada uno de los elementos de la llave K; es decir, tenemos que conocer la distribución de frecuencias relativas de los caracteres que han sido cifrado con cierta $k_i \in K$ y mediante la comparación con la Figura 2.2, adivinar el valor para k_i. La solución al cifrado de Vigenére precisamente consiste en encontrar el período, el inconveniente se traduce en la gran cantidad de esfuerzo manual que hay que invertir.

Dentro de nuestros objetivos se encuentra el de disminuir la labor del criptoanalista para que le sea posible conocer el texto en claro sin mucho esfuerzo, es decir, realizarlo de una forma automatizada mediante un sistema de cómputo.

Los métodos conocidos para el criptoanálisis del algoritmo de Vigenére básicamente son dos: *Índice de Coincidencia* y *Kasiski*. Los explicaremos claramente en la sección 3.2.1 y 3.2.2, pero como comentario introductorio, el criptoanálisis de Kasiski es eficaz para encontrar la solución a un texto cifrado sin importar qué tan grande sea D, aunque requiere de largas horas, mientras que el criptoanálisis por el Índice de coincidencia incorpora un procedimiento sistemático, no muy eficaz cuando el período es muy grande.

Nos enfocaremos principalmente en el Índice de Coincidencia con la finalidad de encontrar las condiciones necesarias para aumentar la eficacia en su uso, automatizar el procedimiento mediante un sistema de cómputo y además también automatizaremos los siguientes pasos a seguir una vez que se ha conocido el período, ya que tanto el criptoanálisis de Kasiski e Índice de coincidencia no contemplan un método para conocer el contenido de la llave K.

3.2.1 Kasiski

EL cifrado de Vigenére durante 300 años se consideró perfecto, hasta que en 1863 un militar Prusiano de nombre *Kasiski* propuso un método para romperlo. Su método consistió en la búsqueda del tamaño de la llave D.

Sabemos, así lo indicamos en la sección 3.2, que si el tamaño de llave es conocido entonces necesitaremos fragmentar el mensaje dependiendo del número de elementos de la llave y cada partición se analizará como si fuese un texto cifrado producto de sustitución monoalfabética.

El método de Kasiski para encontrar el tamaño de la llave comienza por determinar la distancia entre palabras repetidas en el texto cifrado. Para facilitar la explicación cifraremos un mensaje de 5,000 caracteres, cuyas primeras líneas son:

```
EDUCA   RCOMP   ROMIS   OINEL   UDIBL
EDEQU   IENES   LOGRA   NLOSM   ASALT
OSNIV   ELESA   CADEM   ICOSE   UGENI
AHERR   ERAVO   CEROF   RANCI   SCOBA
```

Mediante la llave:

$$\begin{aligned} K &= \{k_1, k_2, k_3, k_4, k_5, k_6\} \\ &= \{\text{ DILEMA }\} \\ &= \{3, 8, 11, 4, 12, 0\} \end{aligned}$$

El texto cifrado obtenido mediante el cifrado de Vigenére es:

```
HLFGM   RFWXT   DOPQD   SUNHT   FHUBO
MOICU   LMYIE   LROCE   ZLRAX   EEAOB
ZWZIY   MWIEA   FIOIY   IFWDI   GGHVT
ETEUZ   PVMVR   KPVAF   UIYGU   SFWME
```

Es indispensable conocer la posiciones de cada uno de los caracteres de esta manera:

Posición:	1	2	3	4	5	6	7	8	9	10	11	12
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Llave:	D	I	L	E	M	A	D	I	L	E	M	A
Texto Claro:	E	D	U	C	A	R	C	O	M	P	R	O
Texto Cifrado:	H	L	F	G	M	R	F	W	X	T	D	O

Posición:	13	14	15	16	17	18	19	20	21	22	23	24
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Llave:	D	I	L	E	M	A	D	I	L	E	M	A
Texto Claro:	M	I	S	O	I	N	E	L	U	D	I	B
Texto Cifrado:	P	Q	D	S	U	N	H	T	F	H	U	B

Cuando se ha obtenido el texto cifrado de alguna manera, es importante verificar que el texto cifrado no sea producto de una sustitución monoalfabética, para ello contraponemos la Figura 3.4 con nuestro gráfica de referencia (Figura 2.2), de donde concluimos que un corrimiento no es la diferencia entre ambos.

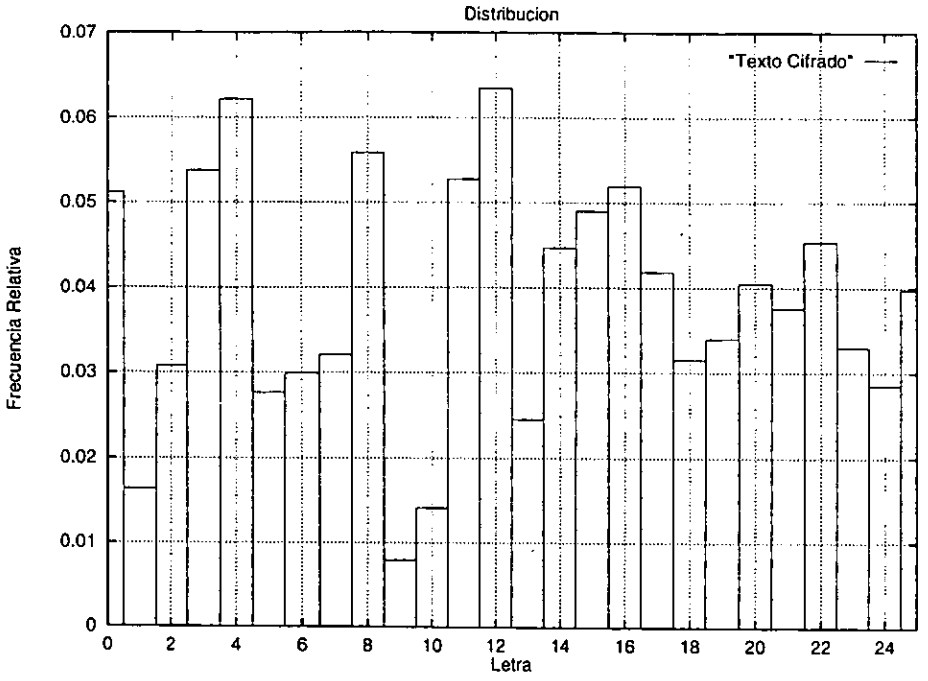


Figura 3.4: Distribución de Texto Cifrado

Ya que se ha descartado la posiblidad de que el texto cifrado no es producto de una sustistución monoalfabética, utilizando el texto cifrado de nuestro ejemplo, observamos que en la posición 7 y 8 se encuentran las letras "FW" las que después aparecen en la posición 67-68. También tenemos a las letras "EA" en la posición 47-48 y 59-60. Se tendrá un mayor éxito entre un mayor número de observaciones, sin embargo con las que hemos realizado es suficiente para una buena aproximación.

La distancia de separación entre un "FW" y otro es de 60 caracteres, mientras que para "EA" es de 12. El que coincidan éstas letras implica que la llave utilizada ha cifrado los mismo caracteres. Aunque no sabemos el tamaño preciso de la llave, sabemos que ha de ser un factor o múltiplo de 60 y 12.

Es decir, las estimaciones para el período es de:

$$D = \{2, 3, 6, 12, \ldots, 60, 120 \ldots\}.$$

Es aquí donde termina el criptoanálisis de Kasiski, el siguiente paso es el de probar cada uno de los valores del tamaño de llave encontrado hasta que obtengamos un texto en claro.

Comenzando con $D = 2$ elaboraremos un diagrama de distribución de frecuencias suponiendo que se utilizaron dos corrimientos. Es decir, si es correcta nuestra suposición significa que $K = \{k_1, k_2\}$ y al graficar las frecuencias relativas de todas las letras cifradas con k_1, deberá ser un diagrama similar al de nuestro patrón de referencia.

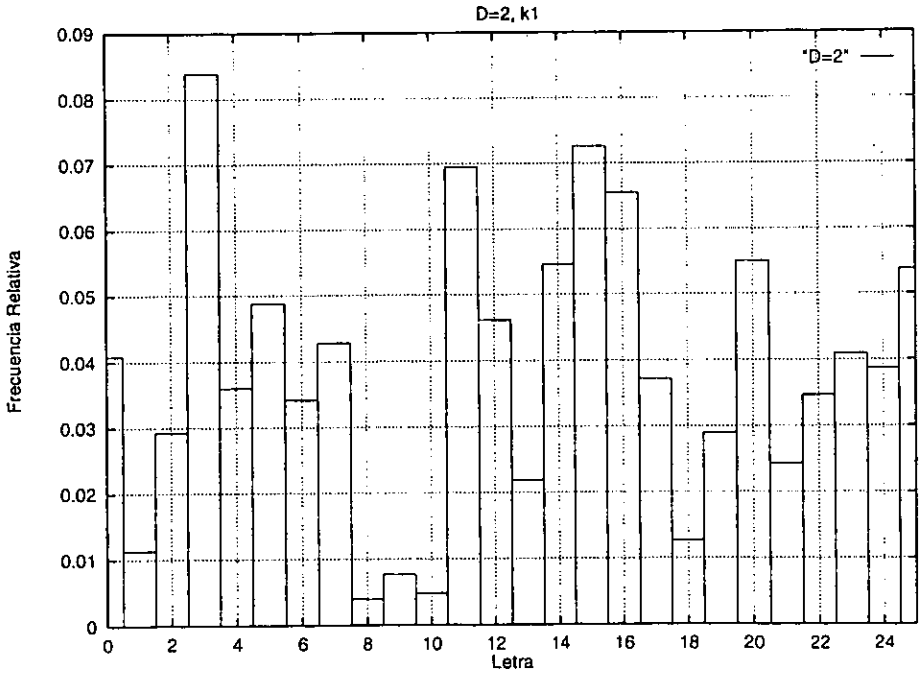


Figura 3.5: *D=2* , k_1

Sin embargo en la Figura 3.5 observamos que la distribución de los caracteres cifrados con k_1 no es similar a la Figura 2.2; con lo que tenemos la certeza de que el período empleado es incorrecto. Lo que obliga a investigar el comportamiento cuando $D = 3$.

Análogamente obtenemos una distribución 3.6 para aquellas letras cifradas con k_1, que corresponde al primer elemento de los tres que se encuentran en la llave K. Observamos que la Figura 3.6 no se ajusta a la deseada (Figura 2.2). Hay que puntualizar que no hay una ventaja en estudiar a las letras cifradas con k_1, k_2 o k_3; independientemente de la selección realizada, de ser correcta el tamaño de llave el diagrama final tendría un comportamiento equivalente al de la Figura 2.2.

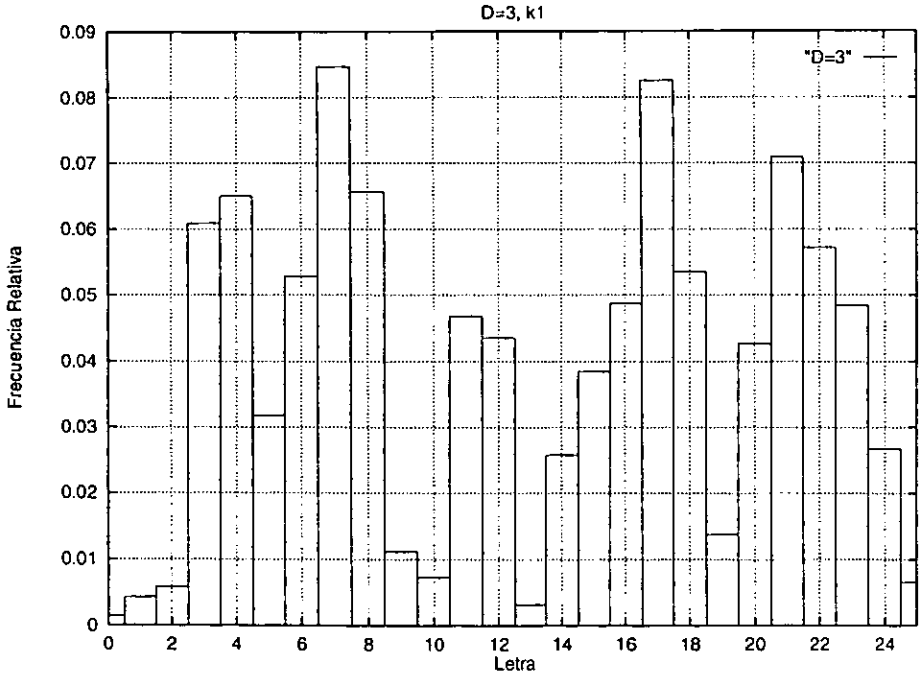


Figura 3.6: $D = 3$, k_1

Tampoco existe una razón lógica para iniciar nuestra búsqueda mediante $D = 2$, en ocasiones se tiene la suerte de escoger en el primer intento el tamaño de llave adecuado utilizando otro orden de selección.

Empleando $D = 6$ y k_1: la Figura 3.7 se asemeja en mucho a la anhelada. También se induce que el corrimiento utilizado para k_1 es de 3 unidades.

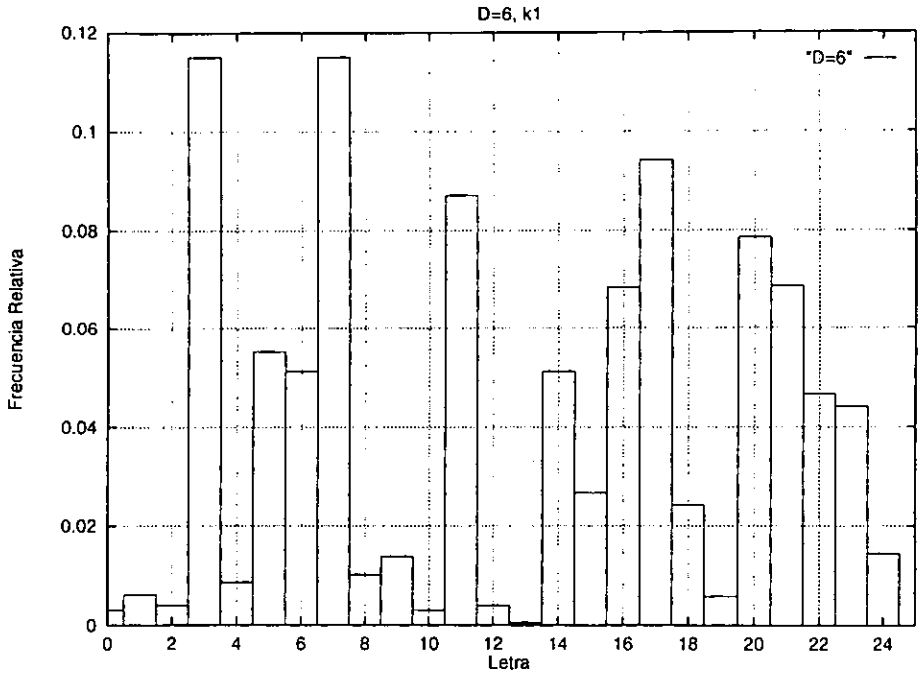


Figura 3.7: $D = 6$, k_1

El trabajo siguiente consiste en realizar una gráfica similar a la de la Figura 3.7 pero para k_2, k_3, k_4, k_5 y k_6. Para conocer el corrimiento asociado a cada uno de éstas y con ello conocer el valor de la llave K empleada para cifrar el texto anterior.

El criptoanálisis de kasiski, que hemos ilustrado, proporciona una aproximación del tamaño de la llave, y aunque ya tengamos una forma de atacar el cifrado de Vigenére, también observamos que es laborioso encontrar el período al igual que el contenido de la llave K.

Con el Índice de Coincidencia mostraremos otra alternativa basada en matemáticas que simplifica totalmente el trabajo que hemos realizado.

3.2.2 Índice de Coincidencia

El índice de coincidencia (IC) basa su método en la comparación de las frecuencias relativas de un texto cifrado y la de una distribución uniforme (Figura 3.8).

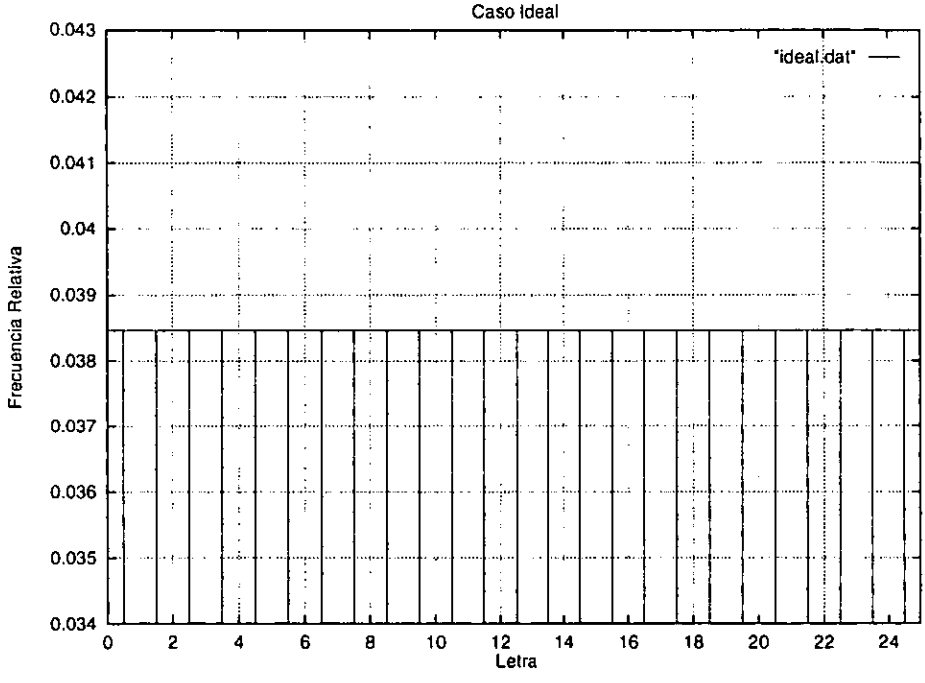


Figura 3.8: Distribución Uniforme

Para describir la Figura 3.8, la cual llamaremos *caso ideal*, requerimos considerar que la frecuencia de cada letra sea la misma, es decir:

frecuencia de "A"=frecuencia de "B"=...=frecuencia de "Z"

que lo simbolizaremos como:

$$f_A = f_B = \ldots = f_Z$$

Lo que implica que tanto la frecuencia relativa como la probabilidad estén definidos como:

$$p_i = \frac{f_i}{N} \quad i = A, \ldots, Z$$

Donde N: es el *tamaño del texto*

y puesto que $\sum_{i=A}^{Z} p_i = 1$, entonces

$$p_i = \frac{1}{26} \quad \text{, donde } i = A, \ldots, Z$$

Ésto no sucede en la vida real, como lo dijimos en la sección 2.1. por lo tanto nos referiremos a la Figura 2.2 como la situación real. Por lo tanto, nos podemos cuestionar sobre la diferencia entre la probabilidad real y la del caso ideal, es decir:

$$p_i - \frac{1}{26} \quad \text{, donde } i = A, \ldots, Z$$

Como la variación de cada una de las probabilidades tiene que estar en función de las 26 cantidades, sumamos cada una de las contribuciones, por lo que se propone:

$$\sum_{i=A}^{Z} (p_i - \frac{1}{26})$$

El inconveniente con la propuesta anterior es que carece de información útil, porque:

$$\sum_{i=A}^{Z} (p_i - \frac{1}{26}) = \sum_{i=A}^{Z} p_i - \sum_{i=A}^{Z} \frac{1}{26} = 1 - 26(\frac{1}{26}) = 0$$

Para evitar este problema es factible considerar la suma de los cuadrados de la diferencia de cada cantidad, i.e

$$\sum_{i=A}^{Z} (p_i - \frac{1}{26})^2$$

La fórmula anterior lleva el nombre de *Measure of roughness* (MR) en Inglés o *medida de aspereza* [ROB82]. Desarrollando dicha expresión observamos que:

$$\begin{aligned} MR = \sum_{i=A}^{Z}(p_i - \frac{1}{26})^2 &= \sum_{i=A}^{Z}(p_i^2 - \frac{2}{26}p_i + \frac{1}{(26)^2}) \\ &= \sum_{i=A}^{Z} p_i^2 - \frac{2}{26} + 26(\frac{1}{(26)^2}) \\ &= \sum_{i=A}^{Z} p_i^2 - \frac{1}{26} \\ &\approx \sum_{i=A}^{Z} p_i^2 - 0.038 \end{aligned}$$

y finalmente: $\sum_{i=A}^{Z} p_i^2 \approx MR + 0.038$.

De conocer p_i definimos MR, sin embargo en un texto cifrado no conocemos dicha cantidad y más aún no conocemos el significado de la probabilidad. Por lo tanto es indispensable hallar una aproximación. Sabiendo que p_i^2 es la probabilidad de que al seleccionar al azar dos letras, en un texto, éstas sean iguales, aplicamos el siguiente resultado [ROB82]:

$$\begin{pmatrix} N \\ 2 \end{pmatrix} = \frac{N(N-1)}{2} \quad \# \textit{ maneras de tomar 2 letras al azar}$$
$$\begin{pmatrix} f_i \\ 2 \end{pmatrix} = \frac{f_i(f_i-1)}{2} \quad \# \textit{ maneras de tomar 2 letras iguales}$$

Así, la probabilidad de que al tomar dos letras al azar, éstas sean iguales es:

$$IC = \frac{\sum_{i=A}^{Z} f_i(f_i-1)}{N(N-1)}$$

La expresión anterior se le conoce como *índice de coincidencia* (IC). Este número es una estimación de $\sum_{i=A}^{Z} p_i^2$, y por tanto también lo es de $MR+0.038$. Y a diferencia del MR, se puede obtener del texto cifrado.

Con los resultados anteriores, el criptoanalista al conocer el cifrado de Vigenére, está en la libertad de emplear alguna K de cierta longitud

y determinar cuál es el índice de coincidencia asociado. Cuando reciba en sus manos algún texto cifrado le restará calcular el IC y relacionarlo con el período que él ha calculado previamente. Por ejemplo, en la literatura consultada [ROB82, SIN66, STI95], se muestran los siguientes resultados, para el idioma inglés, del índice de coincidencia y el período de llave correspondiente.

D	IC
1	0.066
2	0.052
3	0.047
4	0.045
5	0.044
10	0.041
mayor	0.038

A continuación presentamos la tabla correspondiente, para el español, obtenida del análisis experimental sobre textos en Español [1].

D	IC	[2]σ	[IC-σ , IC+ σ]
1	0.0747	0.0014	[0.0733, 0.0761]
2	0.0556	0.0050	[0.0506, 0.0606]
3	0.0494	0.0037	[0.0457, 0.0531]
4	0.0459	0.0028	[0.0431, 0.0487]
5	0.0440	0.0026	[0.0414, 0.0466]
6	0.0429	0.0027	[0.0402, 0.0456]
7	0.0420	0.0019	[0.0401, 0.0439]
8	0.0415	0.0018	[0.0397, 0.0433]
9	0.0411	0.0013	[0.0398, 0.0424]
10	0.0407	0.0014	[0.0393, 0.0421]
mayor	0.037	–	–

Para ilustrar como se aplica el índice de coincidencia, haremos el análisis sobre un texto cifrado en español de 5,000 caracteres.

[1] El procedimiento para encontrar dichos valores se incluye en el anexo

[2] σ denota la *desviación estándar*, la cual mide la dispersión de los datos respecto a la media.

Supongamos que hemos recibido el texto cifrado mediante el cifrado de Vigenére, cuyas primeras lineas anexamos y sabemos que el mensaje original estaba escrito en español:

```
BGYRX VNPYK XVDVI LTPDG NCMYÑ VOLÑI KOPYM DIMAZ ZLMDU
IGEQK HKÑFU LTAPK ÑCOCX ZUYRI OGWRI KÑGAÑ XCÑVU IGERQ
YGEHÑ IQÑCS EWYHU GQYQX ZUORR VAAOQ KQXOK NIRRS ZTMYK
GGÑHX DEÑCV GEWÑI KÑBÑT DCNFÑ OCYVI VFPQK AGYGG MWPGK
OTMAY AQDZG ZOQÑH NKÑÑS OGORK MWTDU OGWRL KOTPU ÑGPAI
PGYHX VGYDQ VVTPG ÑRMFG AQDZG NWYÑG GKMAF VEAAS ZEÑCX
LNMZG TQDSG WTTPG IVPWG LQYRY VFPZÑ XXTAP NDREM IKÑBI
ZVFAF GÑRPF YKOMY KÑKYS UNÑAI SVRAF ZVXAN JZOPP.....
```

Al determinarl el índice de coincidencia, obtenemos IC=0.0434; que en el caso del español corresponde a una llave de tamaño 5, 6 ó 7. Conociendo dicha información, se lleva a cabo un análisis estadístico de sustitución monoalfabética para cada uno de los alfabetos empleados, utilizando $D = 5, 6, 7$.

De ésta manera, al escoger el período adecuado obtenemos las distribuciones de la Figura. 3.9; que representan el corrimiento utilizado. Al comparar con la distribución de un texto en claro, será posible obtener tal corrimiento. En cada una de las gráficas de la Fig. 3.9, se han resaltado las frecuencias de las letras 'A' y 'E', denotados con los valores 0 y 4 respectivamente.

Considerando la gráfica del alfabeto 1, dichas frecuencias corresponden a las letras 'V' y 'Z', con valores 22 y 26; así el corrimiento entre 'A' y 'V' es de 22, entre 'E' y 'Z' también, y lo mismo ocurre con las demás letras. Este procedimiento se repite para cada uno de los alfabetos.

Como hemos usado un alfabeto de 27 letras, en donde se incluye a la ñ, es así como concluimos que la llave usada es $K = \{22, 2, 12, 14, 6\}$; que corresponde a la palabra VCMÑG. Al descifrar el mensaje e incorporando puntuación y espacios adecuados obtenemos:

GENERAL ELECTRIC PREPARA ALIANZA CON EL GIGANTE JAPONES DE MICROPROCESADORES NEC TELECOMUNICACIONES EL DESTINO CONJUNTO LONDRES, DE MAYO BLOOMBERG. GENERAL ELECTRIC, CO, PLC, LA COMPAÑIA BRITANICA DE DEFENSA QUE SE TRANSFORMA EN FABRICANTE DE EQUIPO TELEFONICO, SE ENCUENTRA EN PLATICAS PARA FORMAR UNA ALIANZA CON NEC, CORP, LA MAYOR FABRICANTE JAPONESA DE MICROCHIPS Y COMPUTADORAS PERSONALES, INFORMO UNA PORTAVOZ DE NEC...

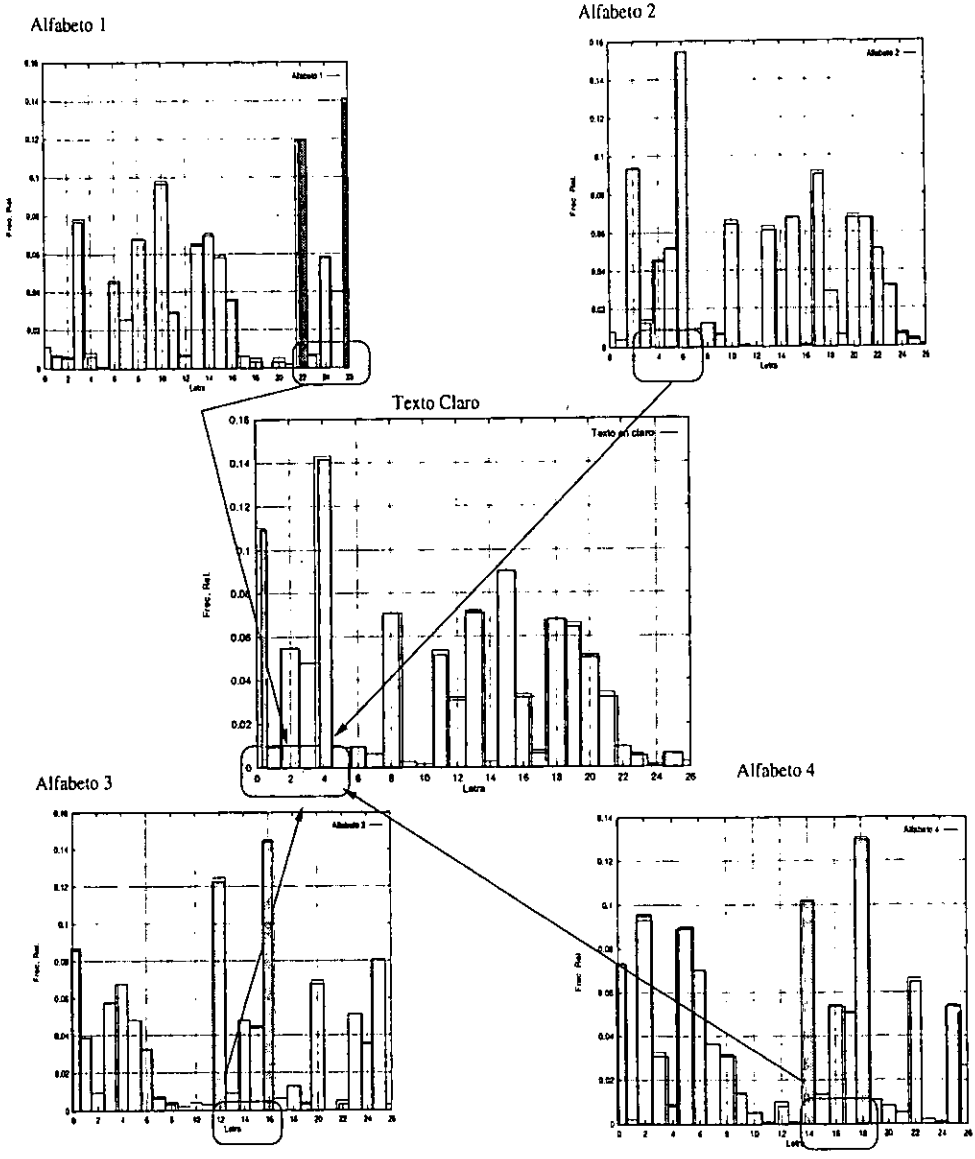


Figura 3.9: Distribución de frecuencias, de un texto claro en español y de cuatro de los cinco alfabetos.

3.3 Implementación

El desarrollo que aquí incluimos lo explicaremos en diversas etapas, la primera consiste en resolver el problema asociado a la estimación del tamaño de llave desconocido.

Nosotros consideramos como elemento sustancial para encontrar el período el índice de coincidencia, por lo que también aquí anexamos la rutina necesaria para encontrar dicho valor.

La siguiente etapa consiste en determinar el corrimiento correspondiente a cada uno de los elementos de la llave involucrada. Mediante estos dos pasos, lo que se obtiene es el texto en claro asociado al texto cifrado, cuando no se conoce la llave involucrada.

En esta sección nos enfocaremos a las rutinas encargadas en realizar las dos etapas, que hemos mencionado, que son de nuestro principal interés. Sin embargo hay que aclarar que también se desarrollaron una serie de programas cuya finalidad fué la de evaluar cada una de nuestras propuestas.

Fué importante para nuestro trabajo relacionar cada uno de nuestros resultados para obtener como producto final un programa, incluido en el Anexo A.3, cuyo objetivo es el descifrar un mensaje sin el conocimiento de la llave.

Cada uno de los programas que incluimos se encuentran en el Lenguaje de programación ANSI C los cuales fueron compilados y probados en la supercomputadora Origin 2000. Aunque la Supercomputadora Origin 2000 cuenta con 40 procesadores, éstos se encuentran agrupados en 2 módulos, uno con 8 procesadores y otro con 32. Nuestros programas fueron compilados y probados el el módulo correspondiente a los 8 procesadores, lo cual fue más que suficiente.

Gracias al equipo de cómputo utilizado fué posible realizar diversas pruebas en un tiempo muy corto y gracias a ello obtuvimos los resultados que hemos incluido en las conclusiones.

Una vez relizado éstas aclaraciones nuestra siguiente tarea es la de mostrar cada una de las funciones que son cruciales en nuestro desarrollo.

3.3.1 Índice de Coincidencia

Para determinar el índice de coincidencia simplemente se empleó la fórmula presentada en la sección 3.2.2.

La consideración adicional que realizamos en el presente trabajo es la incorporación del argumento nombrado *tope*, que determina el número de caracteres con los que que cuenta el alfabeto considerado.

Por ejemplo, para verificar los datos incluidos en la diversa literatura, fué necesario considerar $tope = 26$, sin embargo para el cálculo del índice de coincidencia del español hemos incluido un caracter más, la letra ñ, por lo que $tope = 27$.

Con el razonamiento que hemos utilizado, si se desea conocer el índice de coincidencia asociado a otros alfabetos, lo único que hay que hacer es definir en la variable *tope* el número de elementos.

Adicionalmente, para definir un alfabeto nuevo, diferente a los que proponemos con 26 y 27 caracteres, es necesario modificar el archivo *abc.c* e *inversa.c*, en donde almacenamos cada uno de los elementos del afabeto con su correspondiente entero.

El código que cumple con la determinación del índice de coincidencia lo hemos incluido:

```
/* DETERMINACION EL INDICE DE COINCIDENCIA.

 Entrada:
char *document: Nombre del archivo que
                        se pretende analizar
int tope      : Numero de elementos
        del alfabeto utilizado.

 Salida:
Indice de coincidencia
*/


float index(char *documento, int tope){
 int alfa[tope], k=0, N=0;
 char a;
 FILE *tempo;
 float valor=0;

 /* Abrir archivo, segun el nombre indicado en *docuumento */
 tempo=fopen(documento,"r");

 /* Comprobar que el si existe y se puede leer archivo */
 if(tempo==NULL){
   printf("Error en archivo (index.c)\n");
   exit(1);
 }

 /* Inicializa vector alfa utilizado para almacenar
    valores de frecuencias absolutas cada uno de las letras */
   for(k=0;k<tope;k++)
     alfa[k]=0;

 /* Lectura del archivo, y contabilizacion de la frecuencia
    relativa de cada letra */
    while (!feof(tempo)){
     if (inversa((a=getc(tempo)), tope) < tope)
```

```
      alfa[inversa(a, tope)]++;
    }

 /* Calcular el tamanio del texto */
    for(k=0;k<tope;k++)
      N+=alfa[k];

 /* Calcular el valor del Indice de coincidencia */
    for(k=0;k<tope;k++)
      valor+=alfa[k]*(alfa[k]-1);
    valor=(valor/(float)(N*(N-1)));

 /* Cerrar archivo de lectura, y regresar el valor
    del indice de coincidencia calculado */
 fclose(tempo);
 return valor;
}
```

3.3.2 Estimación del Período

Para conocer la llave de algún texto cifrado, hemos acordado desde secciones anteriores que uno de los primeros pasos consiste en encontrar el tamaño de la llave.

Nuestro objetivo en esta sección es la de proponer un mejor método que cumpla con dicha tarea, al igual que deseamos que sea de una manera eficiente. Para ello fue necesario desarrollar un programa de cómputo, el cual mostraremos.

Nuestro argumento principal se basa en el índice de coincidencia, aunque sabemos que éste es "confiable" hasta un período menor a 10, nosotros hicimos algunas modificaciones para extender su uso.

En primer lugar, supusimos un período desde $D = 1$ hasta un tercio del número de caracteres, ya que de ser tomar un período mayor, ésto nos llevaría una mala aproximación del índice de coincidencia.

El procedimiento se realizó mediante varias iteraciones, las cuales consistieron en suponer un tamaño de llave, que se relacionó con el número de iteración, y después se calculó a cada uno de los caracteres que fueron cifrados con el mismo k_i su correspondiente índice de coincidencia.

Por ejemplo, en la iteración 3 trabajamos con un $D = 3$ y además calculamos el índice de coincidencia a los caracteres que fueron cifrados con k_1, k_2 y k_3.

Antes de realizar la siguiente iteración calculamos el promedio de los índices de coincidencia, y evaluamos si dicho valor coincide con el índice de coincidencia cuando $D = 1$, de ser cierto entonces hemos encontrado el período y en caso contrario suponemos un tamaño de llave mas grande.

La justificación por la cual utilizamos dicho razonamiento surge del planteamiento inverso: Supongamos que conocemos el período. es decir, sabemos el número de corrimientos que ha sufrido el texto cifrado, por lo tanto, es viable agrupar en un conjunto aquellos caracteres que sufrieron el primer corrimiento, en otro los que se les aplicó el segundo corrimiento, y así susceviamente hasta el último corrimiento.

Como cada elemento de la llave K equivale a un corrimiento simple, significa que si determinamos el diagrama de distribución de los caracteres cifrados con la llave k_i, entonces obtenemos un diagrama similar a la Figura 2.2 con la diferencia de que las barras estarían corridas k veces.

Recordemos que el índice de coincidencia se basa en la diferencia entre la distribución de un texto cifrado y la de una distribución uniforme, es decir, que no importa si las barras estan desplazadas un número de veces, por lo tanto, el conjunto de caracteres al que se les ha aplicado un corrimiento deberán tener un índice de coincidencia al determinado cuando $D = 1$.

Como nosotros no conocemos D, optamos por *suponerlo* desde $D = 1$ hasta un tercio del tamaño del texto, por lo tanto, en cada una de las iteraciones se evalúa que los conjuntos de cada uno de los corrimientos cumplan con el índice de coincidencia cuando $D = 1$.

Dado que el índice de coincidencia cuando $D = 1$ no es exacto nosotros hemos considerado una cota superior y una cota inferior, que dependerá del cálculo previo del índice de coincidencia, lo que significa que si se desea utilizar otro idioma, será necesario actualizar dichos valores. La razón por la cual escogimos dichos intervalos se encuentra justificado en la sección 3.2.2. El código que cumple con éste objetivo lo hemos incluido:

```
/* ESTIMACION DEL PERIODO

Entrada:
char *documento : Nombre del archivo a estudiar
int LETRAS : Numero de elementos que
             tiene el alfabeto que se utilizara

Salida
  Tamanio de llave

NOTA:
El parametro COTAINF y COTASUP, ha sido
previamente calculado, y dichos valores corresponden
al intervalo del indice de coincidencia cuando
el periodo es igual a 1 y el mensaje esta
escrito en espaniol. Si se desea utilizar algun
otro alfabeto y otro idioma, se requerira acutalizar
dicho valor.

*/

/* INICIO ---------------------------------- */
int periodo(char *documento, int LETRAS){
 FILE *fuente;
 double *ICxPeriodo, Suma=0, COTAINF=0.0733, COTASUP=0.0761;
 int *alfabetos, d=1, tope, i, total, j, parar=0,d1;
 char letra;

/* Definir el tamanio maximo de tamanio de
   llave a considerrar */
 tope=txtlen(documento, LETRAS)/3;

/* Abrir archivo fuente, segun nombre de
   indicado en *documento */
 fuente=fopen(documento,"r");

/* Validar el archivo cifrado */
```

```
 if (fuente == NULL){
  printf("Error en archivo (periodo.c)\n");
  exit(1);
 }

/* Hacer hasta que que encuentre el periodo, o
   el periodo sea igual a la variable llamada tope */
  while(parar==0){

  /* Solicitar memoria, para almacenar cada una de las
     frecuencias relativas */
  alfabetos=(int*)malloc((LETRAS*d)*sizeof(int));

  /* Validar la solicitud a memoria */
  if(alfabetos==NULL){
   printf("Error en malloc\n (periodo.c)");
   exit(1);
  }

 /* Inicializar el vector en donde se contaran
    cada una de las frecuencias del texto
    dependiendo del periodo estimado */
 for(i=0;i<(LETRAS*d);i++)
  alfabetos[i]=0;

 /* Realiza el conteo para cada uno de los alfabetos */
  i=0;
  rewind(fuente);
  while(!feof(fuente)){
   letra=getc(fuente);
   if (inversa(letra,LETRAS) <  LETRAS){
    alfabetos[inversa(letra,LETRAS) + LETRAS*(i%d)]++;
    ++i;
   }
  }

 /* Reserva espacio para calcular el indice de
```

```
    coincidencia a cada uno de los caracteres cifrados
    con el mismo corrimiento */
  ICxPeriodo=(double*)malloc(d*sizeof(double));

 /* Validar solicitud de memoria */
  if(ICxPeriodo==NULL){
   printf("Error en malloc\n (periodo.c)");
   exit(1);
  }

 /* Incializa dicho vector a cero */
  for(i=0;i<d;i++)
   ICxPeriodo[i]=0;

 /* Determina IC: */
  for(i=0;i<d;i++){

   Suma=0;
   for(j=0;j<LETRAS;j++)
    Suma+=alfabetos[j + LETRAS*i]*(alfabetos[j + LETRAS*i] - 1);

  /* Determinar las letras asociadas a cada alfabeto */
   total=0;
   for(j=i*LETRAS;j<(i*LETRAS + LETRAS); j++)
    total+=alfabetos[j];

  /* Determinar IC, asociado a dicho periodo */
   if(total >0 && (total -1) > 0){
    ICxPeriodo[i]= Suma/(total*(total-1));
   }else{
    ICxPeriodo[i]=0;
   }

  /* Termina el calculo de IC para todos el i-esimos alfabetos */
}

/* Calcular el promedio de los IC calculados */
```

```
 Suma=0;
 for(i=0;i<d;i++)
  Suma+=ICxPeriodo[i];
 Suma/=d;

/* Verificar que se encuentra en el intervalo deseado*/
  if((Suma>=COTAINF && Suma <= COTASUP) || d>=tope){
   parar=-1;
   d1=d;
  }
/* Si se ha encontrado el periodo termina, de lo contrario
   se hace otra iteracion */

/* Se libera la memoria que se ha solicitado para realizar la iteracion*,
  free(alfabetos);
  free(ICxPeriodo);
}

* Fin de la funcion: Se cierra el archivo que se ha analizado,
  y se regresa el tamanio de llave que se ha calculado */
close(fuente);
eturn d1;
```

3.3.3 Estimación del Corrimiento

En la determinación del corrimiento, para cada uno de los elementos de la llave K de Vigenére, hemos desarrollado un programa que no necesita que un criptoanalista realice un diagrama de distribución de frecuencias relativas y realice una gran variedad de pruebas hasta que encuentre un texto en claro.

Para que nuestro método funcione es necesario que el criptoanalista conozca previamente la frecuencia relativa de los caracteres involucrados en el mensaje en claro desconocido, por lo tanto, deberá conocer el idioma en el que se encuentre el mensaje.

Para nuestro trabajo nosotros utilizaremos las frecuencias relativas que hemos ilustrado en la Figura 2.2, por lo que consideraremos exclusivamente textos en claro en español.

El programa que desarrollamos una vez que conoce las frecuencias relativas a utilizar como base, realizara posteriormente 26 iteraciones. En cada una se considerará un corrimiento, que dependrá del número de la iteración, por ejemplo, en la iteración 4 el corrimiento a suponer será de $k_i = 4$.

Una vez que al conjunto de caracteres correspondientes a un corrmiento particular, se descifra con el corrimiento estimado, posteriormente se calcula las nuevas frecuencias relativas, es decir, f_i correponderá a la frecuencia de i y f_i' la frecuencia i' después de aplicar un corrimiento k_i.

Antes de pasar a la siguiente iteración determinamos la suma del producto de frecuenicas f_i con las f_i', almacenamos dicho valor y y realizamos otra iteración.

Para almacenar cada uno de los valores será conveniente auxiliarnos de un arreglo, que lo notaremos como $Vec[]$. Con lo fundado anteriormente, para almacenar la iteración k, tenemos que:

$$Vec[k] = \sum_{i=A}^{Z} (f_i)(f_i')$$

Hay que observar que arreglo Vec contiene 26 elementos, uno para cada iteración realizada. Ya para finalizar, solo queda determinar cuál es el corrimiento correcto y para hacerlo, buscamos la posición del valor máximo del arreglo $Vec[]$.

Para ilustrar lo anterior,supongamos que determinamos el siguiente arreglo con 26 elementos :

$Vec[0]$	=	0.04
$Vec[1]$	=	0.2
$Vec[2]$	=	0.001
$Vec[3]$	=	0.04
$Vec[4]$	=	0.04
$Vec[5]$	=	0.02
$Vec[6]$	=	0.001
$Vec[7]$	=	0.04
$Vec[8]$	=	0.04
$Vec[9]$	=	0.002
$Vec[10]$	=	0.001
$Vec[11]$	=	0.004
$Vec[12]$	=	0.004
$Vec[13]$	=	0.002
$Vec[14]$	=	0.001
$Vec[15]$	=	0.004
$Vec[16]$	=	0.001
$Vec[17]$	=	0.094
$Vec[18]$	=	0.04
$Vec[19]$	=	0.002
$Vec[20]$	=	0.001
$Vec[21]$	=	0.094
$Vec[22]$	=	0.001
$Vec[23]$	=	0.04
$Vec[24]$	=	0.04
$Vec[25]$	=	0.002

Si buscamos el valor máximo encontramos que éste se encuentra en la posición 1, por lo tanto el valor de $k_i = 1$.

El método que hemos ilustrado es realizado para todos los elmentos de la llave K, por lo tanto si el período es igual a 5, éste procedimiento se hace 5 veces.

La implementación de la propuesta la hemos incluído en el siguiente programa de cómputo que describe lo mencionado.

```
/* DETERMINAR EL CONTENIDO DE LA LLAVE
   EN BASE AL PERIODO ESTIMADO

 Entrada:
  char *archivo: Nombre de archivo a estudiar
int LETRAS   : Numero de elementos del alfabeto utilizado
int *llave   : vector que corresponde a la
       llave de Vigenere. (el vector no tiene
       valores almacenados)
int d        : Numero de elementos de la llave de
       Vigenere.
 Nota:
Se requiere que exista un archivo llamado "base.dat",
en donde se encuentren las frecuencias relativas para
cada una de las letras del alfabeto utilizado en un
texto en claro.

 Salida:
  El vector de entrada llamado *llave, tiene en cada
uno de sus componentes los valores correspondiente
a la llave de Vigenere desconocida.
*/

/* Funcion que ordena de mayor a menor un vector, indicamos
   la funcion prototipo a utilizar */
void sort2(float*, int);

void corrimiento (char *archivo, int LETRAS, int *llave, int d){
  float sumas[LETRAS],alfabetos[LETRAS*d], tempo[LETRAS],base[LETRAS],d;
```

```
 float totales[d];
 int i,j,k;
 char letra;
 FILE *fuente, *BD;

/* Abrir archivo cifrado */
 fuente=fopen(archivo, "r");

/* Validar que dicho archivo exista y se puda leer */
if(fuente == NULL){
  printf("Error en archivo (corrimiento.c)");
  exit(0);
 }

/* Abrir archivo que contenga las frecuencias de base */
BD=fopen("base.dat","r");
if(BD==NULL){
  printf("Error en archivo base.dat\n");
  exit(0);
}

/* Incializa vector de correlaciones a cero */
 for(j=0;j<LETRAS;j++){
  sumas[j]=0.0;
  base[j]=0.0;
 }

/* Inicializar el vector en donde se contararan
   cada una de las frecuencias del texto
   para cada del periodo estimado */
for(j=0;j<(LETRAS*d);j++){
 alfabetos[j]=0.0;
}

/* Lectura de datos */
 j=0;
 dato=0;
```

```
 rewind(BD);
 while(!feof(BD)){
  fscanf(BD,"%f",&dato);
  base[j++]=dato;
 }

/* Realiza el conteo para cada uno de los alfabetos */
 j=0;
 rewind(fuente);
 while(!feof(fuente)){
  letra=getc(fuente);
  k=inversa(letra,LETRAS);
  if (k <  LETRAS){
   alfabetos[inversa(letra,LETRAS) + LETRAS*(j%d)]++;
   ++j;
  }
 }

/* Cerrar archivos utilizados */
 fclose(fuente);
 fclose(BD);

/* Almacena frecuencia absolutas por periodo */
 for(k=0;k<d;k++){
  totales[k]=0.0;
 }

/* Almacena las distribuciones relativas de cada iteracion */
  for(k=0;k<d;k++){
   for(i=0;i<LETRAS;i++){
     totales[k]+=alfabetos[i + LETRAS*k];
   }
   for(i=0;i<LETRAS;i++){
     if(totales[k]==0){
      alfabetos[i + LETRAS*k]=0;
     }else{
      alfabetos[i + LETRAS*k]/=totales[k];
```

```
      }
    }
  }

/* Encontrar el corrimiento de cada elemento de la llave*/
 for(i=0;i<d;i++){
   for(k=0;k<LETRAS;k++){
      for(j=0;j<LETRAS;j++){
        sumas[k]+=alfabetos[(j +k)%LETRAS + LETRAS*i]*(base[j]);
      }
    }

  /* Hacer una copia temporal */
  for(j=0;j<LETRAS;j++){
   tempo[j]=sumas[j];
  }

  /* Ordena el vector temporal */
   sort2(tempo,LETRAS);

  /* Busca la posicion del valor maximo */
   for(k=0;k<LETRAS;k++){
    if(sumas[k]==tempo[LETRAS-1]){
     j=k;
    }
   }

  /* Determina el contenido de la llave en su
     posicion i-esima */
  llave[i]=(-1)*j;

 /* Inicializa el valor de sumas */
   for(k=0;k<LETRAS;k++){
    sumas[k]=0.0;
   }
 }
```

```
/* Fin de encontrar periodo */
}

/* Funcion de ordenacion */
 void sort2(float A[], int N ){
  int i, j, In=0;
  float Tmp;

  for(In=N/2;In>0;In/=2)
   for(i=In;i<N;i++){
    Tmp=A[i];
    for(j=i;j>=In;j-=In)
     if(Tmp < A[j-In])
      A[j]=A[j-In];
     else
      break;
     A[j] = Tmp;
    }
  }
```

Capítulo 4

Conclusiones

Las matemáticas han sido sustanciales para el desarrollo de cada uno de los capítulos, es importante señalar que gracias a éstas hemos sido capaces de realizar el criptoanálisis del cifrado de Vigenére e introducir las matemáticas involucradas en la criptografía de llave pública.

Hablando del criptoanálisis que realizamos al cifrado de Vigenére es importante señalar que no es el único camino a seguir, pero sin duda en comparación con otros, que incluyen una gran cantidad de trabajo manual, el nuestro propone una alternativa donde el trabajo involucrado por el criptoanalista disminuye considerablemente.

Nos da gusto haber cumplido nuestros objetivos planteados:

1. Comprender los fundamentos básicos de la la criptografía de llave secreta y pública.
2. Extender los resultados del criptoanalisis del cifrado de Vigenére mediante el desarrollo de un sistema de cómputo.

Aunque es cierto que explicar algunos conceptos matemáticos en ocasiones es una labor complicada, esperamos haber ilustrado la importancia que juegan en la criptografía, ya que sin ellas sería prácticamente imposible lograr que los criptosistemas sean empleados por una gran variedad de organizaciones hoy en día.

La justificación para aseverar que perfeccionamos el criptoanálisis del cifrado de Vigenére se fundamentó en la elaboración de una serie de programas de cómputo, incluidos en los Anexos, que evaluaron nuestra propuesta explicada a detalle durante el capítulo 3.

En el presente trabajo, de manera concreta, llegamos a las siguientes conclusiones, las cuales explicaremos a detalle en las secciones siguientes:

- **El criptoanálisis del cifrado de Vigenére no se puede automatizar al 100 por ciento**: El trabajo del criptoanálisis, como lo indicamos en la sección 1.8, requiere de mucha creatividad y aunque hemos automatizado gran parte de las tareas que intervienen ésto no significa que para cualquier texto cifrado, obtenido a partir del cifrado de Vigenére, encontremos inmediatamente el texto en claro correspondiente.

- **Contemplando las limitaciones estadísticas, hemos determinado exitosamente tanto el período y el contenido de la llave K utilizada en el cifrado de Vigenére**: De las pruebas realizadas encontramos datos satisfactorios en cuanto al éxito de la automatización del descifrado sin el conocimiento de la llave, porque en el mejor de los casos el texto cifrado fue descifrado totalmente y en el peor se obtuvieron aproximaciones del tamaño de la llave involucrada.

- **El equipo de cómputo es escencial para el criptoanálisis realizado**: Se aprovechó el equipo de cómputo para obtener resultados que fueran confiables, de no utilizar un equipo de cómputo hubiera sido necesario emplear los mecanismos manuales, los cuales son muy tardados y ésto hubiera limitado el número de pruebas.

4.1 Automatización del Criptonálisis

Mediante diversas pruebas realizadas a varios textos cifrados conocidos. ineludiblemente fué necesario suponer algunas condiciones. entre las que encontramos:

1. *Conocer el idioma del texto en claro desconocido*: Debido a que cada idioma tiene una distribución de caracteres diferentes, ésto implica que el índice de coincidencia también sufrirá cambios, es decir, el índice de coincidencia no es igual para cualquier idioma.

2. *Conocer el alfabeto*: Los textos cifrados que se analizaron siempre utilizaron el mismo alfabeto, aunque sabemos que las sustituciones pueden ser diversas para facilitar el diseño del programa de cómputo, consideramos el alfabeto castellano.

3. *Utilizar un período controlado*: Al saber que no consideraríamos un período tan grande como letras del texto cifrado. éste caso se excluyó.

4. *Considerar el caso simple*: En la práctica un texto en claro es cifrado varias veces mediante diversas funciones de cifrado, nosotros nos concentramos en la situación cuando el texto en claro es cifrado una sola vez.

Para el criptoanálisis que hemos desarrollado fué importante definir la cantidad necesaria de texto cifrado, sin que los aspectos estadísticos se vieran afectados, para lo que estudiamos el comportamiento del índice de coincidencia conforme se aumentara el período. En la Figura 4.1 observamos que la variación del índice de coincidencia ya no es relevante cuando se emplea un tamaño de llave de 10.

Cuando se cifra un archivo mediante un período mayor a 10 el diagrama de distribución de cada una de las letras es similar a la de una distribución uniforme, y como el índice de coincidencia precisamente es una aproximación de la diferencia entre la distribución uniforme con la de un texto cifrado, entonces los valores son parecido y por lo tanto el índice de coincidencia ya no variará en mucho.

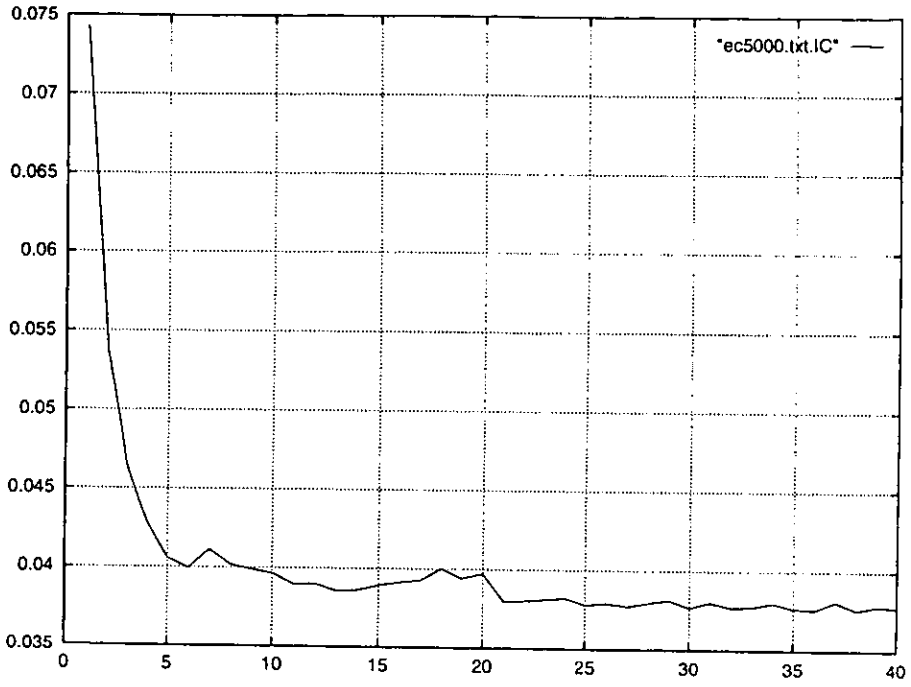


Figura 4.1: Periodo *Vs* índice de coincidencia

Igualmente importante, no solamente el inconveniente se encontró en la poca variación del índice de coincidencia cuando el período era mayor a 10, sino también con la dificultad en determinar diferentes índices de coincidencia para un mismo periodo dado, es decir, como mostramos en la Figura 4.2, observamos que para dos textos cifrados el índice de coincidencia no es el mismo.

Aunque en la diversa literatura consultada se incluye una solo valor para cada índice de coincidencia, después de la observación anterior, consideramos pertinente realizar diversas pruebas para asignar a cada índice de coincidencia un intervalo de valores. Lo cual fue crucial para obtener un mayor éxito en la implementación que determina el período de un texto cifrado.

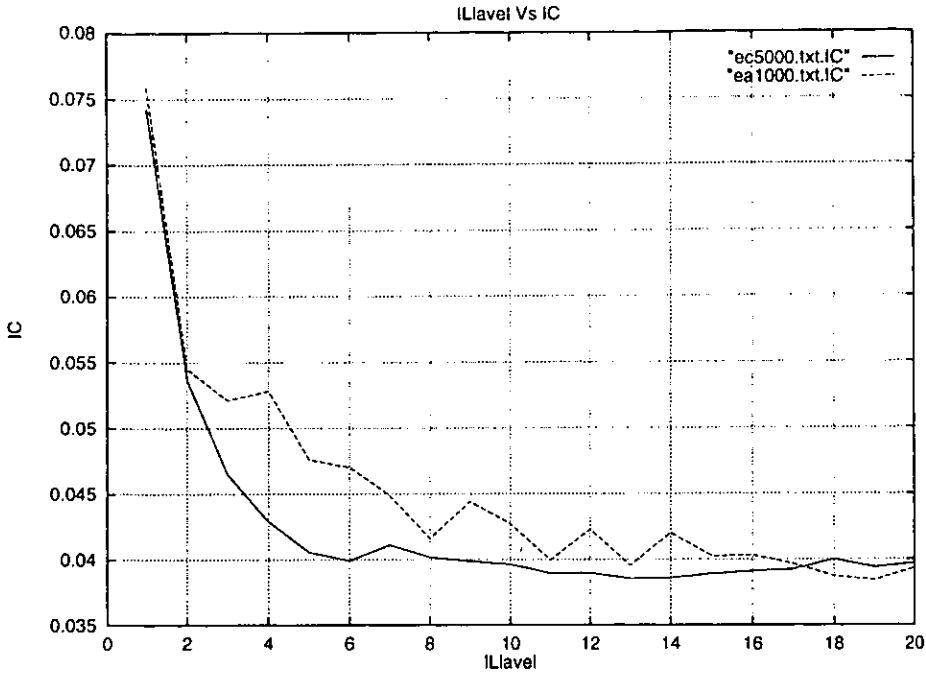


Figura 4.2: Periodo *Vs* IC

Por lo mencionado anteriormente, la naturaleza estadística del índice de coincidencia impide que se elabore un sistema que realice el descifrado automáticamente, también hemos considerado una serie de premisas previas que el criptoanalista requiere.

Finalmente, aunque no sea posible automatizar **todo** el trabajo del criptoanalista hemos simplificado su trabajo por lo que solamente resta interpretar los resultados que se obtengan.

4.2 Determinación del período y corrimiento

Del mecanismo propuesto en la sección 3.3.2 para determinar el período es importante reflexionar sobre los alcances y limitaciones de dicho método.

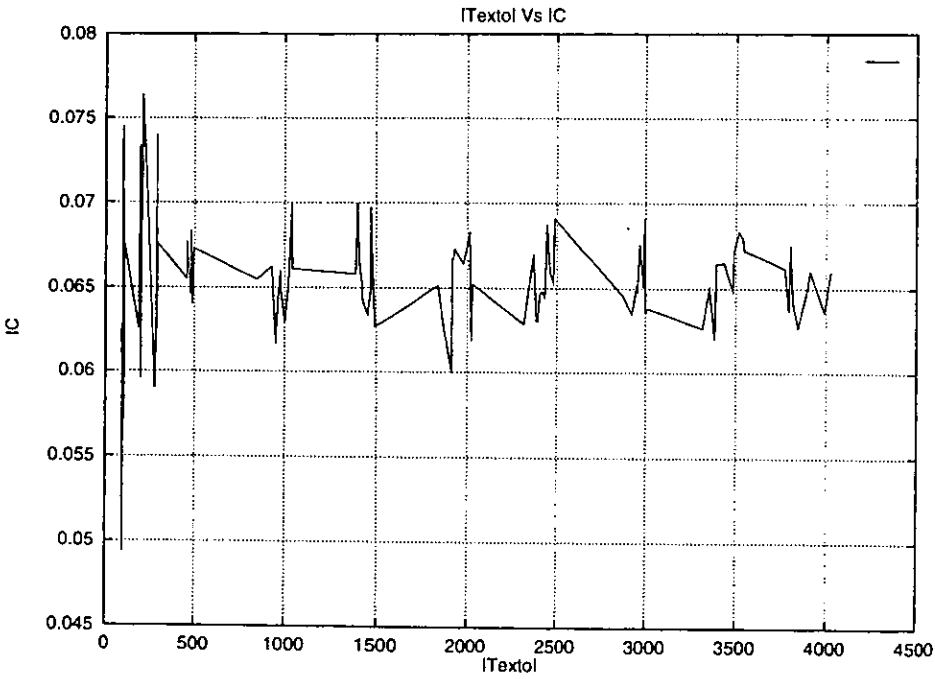


Figura 4.3: Periodo *Vs* IC, con D=1

Para conocer la cantidad de texto cifrado mínimo necesario para encontrar el único texto en claro, observamos el comportamiento del índice de coincidencia cuando el período era igual a 1 y el texto cifrado aumentaba.

Una muestra de nuestras observaciones es la Figura 4.3 en donde se ilustra el efecto que sufre el índice de coincidencia cuando el texto

cifrado varía, con lo que acordamos que era necesario contar con 2000 caracteres para obtener valores confiables.

Una vez que encontramos la cantidad de texto cifrado y los valores del índice de coincidencia correspondiente a cada período, la siguiente etapa consistió en evaluar el método para encontrar el período.

Mediante diversas pruebas encontramos que los errores en el período equivalían a estimaciones erroneas, dado que el período calculado era equivalente a un factor o múltiplo del verdadero período. Aunque sea incorrecta la estimación es una ventaja obtener los posibles valores del del período, al igual que en el método de Kasiski, de un modo más rápido.

Por ejemplo, mediante el cifrado de un archivo mediante diversos tamaños de llaves, al aplicar el método de búsqueda del período encontramos los siguientes errores:

Período Real	Período Estimado	Variación
2058	294	$2058 = (294)(7)$
1278	426	$1278 = (426)(3)$
2175	435	$2175 = (435)(5)$
1320	440	$1320 = (440)(3)$
2205	441	$2205 = (441)(5)$
890	445	$890 = (445)(2)$

Sabemos que una vez encontrado el período la siguiente tarea es encontrar el el valor de cada uno de los elementos de la llave K. dentro de las diversas pruebas encontramos que los errores en definir la llave se relacionaron con la estimación incorrecta del período.

Para ilustrar lo anterior, vamos a anexar el archivo de errores al cifrar un archivo de 5000 caracteres mediante diversas llaves seleccionadas aleatoriamente, considerando un tamaño de llave $D = 1$ hasta $D = 50$ y después aplicar nuestro programa de descifrado automático, tenemos:.

Período Real	Período Estimado	Variación
20	60	$60 = (20)(3)$
40	120	$120 = (40)(3)$
41	246	$246 = (41)(6)$

Los resultados son alentadores porque el trabajo del criptoanalista, a pesar del error del método que empleamos, es simplificado considerablemente, con lo que cumplimos el objetivo concerniente a extender el criptoanálisis del cifrado de Vigenére.

Hay que notar que el emplear el índice de coincidencia, únicamente es confiable hasta un período menor a 10, y como nosotros hemos ilustrado hemos descifrado un periodo igual a 50 con solamente tres errores, lo cual es una adelanto significativo.

4.3 Equipo de cómputo esencial para el criptoanálisis.

Todos los resultados que hemos incluido requirieron de diversas pruebas con una gran variedad de archivos. para procesar cada uno de ellos hubiera sido prácticamente imposibles sin haber contado con un equipo de cómputo.

En primer lugar fue importante tener acceso a Internet para que los archivos seleccionados fueran de los más diversos, es decir. consideramos mensajes de periódicos, poemas, canciones, articulos. Posteriormente en la etapa de procesar cada uno de los archivos se empleó poco tiempo, lo que hizo posible el incorporar los cambios pertinentes para mejorar la eficiencia de los programas hasta obtener los resultados satisfactorios.

Ya con lós resultados obtenidos también fué necesario manipular los datos para elaborar los diversos gráficos, diagramas y algunos cálculos para lo cual gracias al equipo de cómputo simplificamos el trabajo.

Finalmente, de no contar con el equipo de cómputo los resultados presentes no hubiéran sido validados con tanta rigurosidad al igual que hubiéramos tenido que invertir una cantidad de tiempo mucho mayor y el criptoanálisis del Cifrado de Vigenére propuesto sería muy limitado.

Con todo lo discutido y propuesto en el presente trabajo esperamos que los temas hayan sido interesantes y que hayamos ilustrado que las matemáticas y la computación en la actualidad forman parte de nuestra vida y hay que aprender a convivir con ellas.

Bibliografía

[APO92] Apostol Tom. *Calculus Vol 1.* Reverté. 1992.

[CAR90] Cárdenas Humberto, Raggi Francisco. *Álgebra superior* Trillas, 1990.

[CIP87] Deavours Cipher, D. Kahn. *Cryptology, Yesterday, today and Tomorrow.* Artech House, 1987.

[FRA87] Fraleigh John. *Álgebra Abstracta.* Addison-Wesley, 1987.

[KAH67] D. Kahn. *The Codebreakers.* Macmillan, 1967.

[MAS87] Mansuripur Masud. *Introduction to Information theory.* Prentice-Hall, 1987

[PIN93] Pino Caballero *Seguridad Informática. Técnica criptográficas.* Ra-Ma, 1993.

[ROB82] D. E. Robling Denning. *Cryptography and Data Security.* Addison-Wesley, 1982.

[ROS93] Kenneth H. Rossen. *Elementary Number theory and its application.* Addison-Wesley, 1993.

[SIN66] A. Sinkov. *Elementary Cryptoanalysis. A mathematical approach.* Mathematical Association of America, 1966.

[SIN82] Jagjit Sing *Ideas fundamentales sobre teoría de la información, del lenguaje y de la cibernética.* Alianza, 1982.

[SPI92] Michael Spivak. *Calculus.* Reverté, 1992

[STA95] William Stallings. *Network and Internetwork security. Principles and practice.* Prentice Hall, 1995

[STI95] D. R. Stinson. *Cryptography. Theory and Practice.* CRC Press, 1995.

[WIL90] William E. Boyce. *Ecuaciones diferenciales y problemas con valores a la frontera*, Editorial Limusa, México 1990.

[WIJ77] William J. Le Veque *Fundamentas of Number Theory* Dover Publication, 1977

[HAR92] G.H Hardy *An Introduction to the theory of Numbers.* Oxford Science Publication, 1992.

Apéndice A

Apéndice

A.1 Cálculo del índice de coincidencia

Para encontrar el índices de coincidencia dependiendo del período utilizado en un texto cifrado, desarrollamos un programa que cumpliera con dicho objetivo.

Para encontrar un valor confiable, realizamos el cálculo a diversos archivos seleccionados al azar de 500 caracteres hasta 50000 con una variación de 500 caracteres. Cada archivo fue cifrado mediante diversos períodos, los cuales fueron considerados desde $D = 1$ hasta la mitad del número de caracteres del texto y se consideró realizar los siguientes pasos:

1. **Determinación de llave**: El período era conocido sin embargo el contenido de la llave fué seleccionado aleatoriamente
2. **Cifrado del archivo**: Con la llave conocida el archivo era cifrado con la llave definida en el punto anterior.
3. **Cálculo del índice de coincidencia**: Al archivo cifrado se le calculaba el índice de coincidencia.
4. **Almacenar valores**: En un archivo, definido por el usuario, se almacenan el valor del índice de coincidencia calculado y el período conocido.

El programa principal que realiza lo descrito anteriormente lo anexamos a continuación:

```
/* Programa que cifra utilizando VIGENERE
   con una llave aleatoria.

   Se realizan 1 iteracion hasta (Key1) iteraciones,
   en cada una, se calcula el Indice de Coincidencia
   y se almacena su valor en (argv[2]). El archivo
   origen a cifrar se encuentra en (argv[1]).

   La llave utilizada se determina aleatoriamente

OPCIONES:
  TRES=1
    Necesita un tercer nombre de archivo, en el
    que guardara el tama~no de texto y el IC
    cuando el periodo es 1 y 2.

Para desactivar dichas opcion, TRES != 1
*/

#include "alib/criptoanalisis.h"
#define tres 1
#define letras 26

/* Definicion del programa principal */
 void main (int argc, char *argv[]){

 int i=0,k=0,*llave, Key1=0;
 FILE *fuente, *destino, *tempo, *tempo2;
 char a;
 float respuesta=0;

/* Validacion en el numero de argumentos */
 if(tres ==1 && argc != 4){
```

```
  printf("Error en el numero de argumentos \n");
  printf("Hay que introducir: \n ");
  printf("ejecutable 1 2 3   \n");
  printf("donde: \n ");
  printf("1 = Archivo de caracteres de entrada \n ");
  printf("2 = Archivo en donde se almacene el IC \n ");
  printf("3 = Almacen tama~no de Texto y IC cuando d=1,2 \n" );
  exit(1);
 }
  if(tres !=1 && argc !=3){
  printf("Error en el numero de argumentos \n");
  printf("Hay que introducir: \n ");
  printf("ejecutable 1 2   \n");
  printf("donde: \n ");
  printf("1 = Archivo de caracteres de entrada \n ");
  printf("2 = Archivo en donde se almacene el IC \n ");
  exit(1);
 }

/* Abrir y crear archivo fuente y destino */
 fuente=fopen(argv[1],"r");
 destino=fopen(argv[2],"w+");

/* Si existe el tercer archivo, crearlo y validarlo */
 if(tres==1){
  tempo2=fopen(argv[3],"w+");
  if (tempo2==NULL){
    printf("Error en archivo 3");
    exit(1);
  }
 }

/* Validar archivo fuente */
 if(fuente == NULL) {
  printf("Error en archivo origen \n");
  exit(1);
 }
```

```
/* Validar archivo destino */
 if(destino==NULL){
  printf("Error en archivo destino\n");
  exit(1);
 }

/* Definir el tamanio de llave Maximo */
  Key1= txtlen(argv[1],letras)/2;

/* Validar llave utilizada */
  if (Key1 <= 0) {
   printf("Tama~no de llave inv'alido \n);
   exit (1);
  }

/* Cifrar un archivo con una llave aleatoria hasta Key1 */
 for(i=1;i<=Key1;i++){

   /* Abrir archivo temporal en donde se almacena
      el texto cifrado en cada iteracion */
   tempo=fopen("tempo.dat","w+");

   /* Validar archivo temporal */
   if(tempo==NULL){
    printf("Error en iteracion %d\n", i);
    exit(1);
   }

   /* Solicitud de memoria para tama~no de llave
      especificado en la iteracion i-esima */
   llave=(int*)malloc(i*sizeof(int));

  /* Validar tamanio de llave deseado */
   if (llave==NULL){
    printf("Error en llave de longitud %d\n",i);
    exit(1);
```

```
    }

  /* Definir contenido de la llave aleatoriamente */
     fx(&llave[0],i,letras);

 /* Cifrar texto utilizando Vigenere con llave seleccionada */
   k=0;
   rewind(fuente);
   while (!feof(fuente)){
    if (inversa((a=getc(fuente)),letras) >= letras)
     putc(a,tempo);
    else
     putc(abc(cifrado(inversa(a,letras),llave[k++%i],letras),letras),tempo);
   }

 /* Cerrar archivo cifrado  */
  fclose(tempo);

 /* Calcular IC */
  respuesta=index("tempo.dat", letras);

 /* Almacenar en archivo destino, IC correspondiente
     a la iteracion i-esima */
  fprintf(destino,"%d\t %f\n", i, respuesta);

 /* Almacenar en el tercer archivo dado (argv[3])
     el tamanio de texto y el IC, cuando el periodo es uno y dos */
  if(tres==1 && (i==1 || i==2))
   fprintf(tempo2,"%d\t %f\t", txtlen("tempo.dat",letras), respuesta);

 /* Liberar espacio de memoria para la llave i-esima */
  free(llave);

 /* Borrar texto cifrado */
  system("rm tempo.dat");

 /* Repite el proceso, con otro tamanio de llave */
```

```
 }

/* Fin de programa Indices */
 fclose(fuente);
 fclose(destino);
 if(tres==1)
  fclose(tempo2);
}
```

A.2 Evaluación de la estimación del período

Para determinar el período ya hemos descrito el código encargado en hacerlo, sin embargo, para evaluar su alcance consideramos pertinente hacer una evaluación.

Se emplearon archivos seleccionados al azar desde 500 caracteres hasta 50000, con una diferencia entre cada uno de 500 letras y a cada uno de los archivos se les realizó la siguiente prueba:

1. **Cifrar una archivo**: Un archivo, escogido al azar se cifra mediante el cifrado de Vigenére, con una llave aleatoria con período conocido.

2. **Emplear función periodo()**: Al archivo cifrado se le estima el período.

3. **Comparación de resultados**: Si el período real es diferente al estimado entonces almacenamos dichos resultados.

Gracias a los pasos anteriores encontramos el alcance del método propuesto para encontrar el período y el código que cumple con dicho objetivo lo hemos incluido:

```
/* Cifra un archivo dado en (argv[1]) desde 1 hasta tope.
   Si el periodo estimado en (periodo.c) es incorrecto,
   guarda el IC correspondiente y el periodo dado
   en el archivo de errores, denotado en argv[2]. */

#include "alib/criptoanalisis.h"

void main (int argc, char *argv[]){
 int *llave, k, i=0, estimado, letras=26,tope;
 FILE *fuente, *destino, *tempo;
 char a;

/* Validacion en la entrada de datos. */
 if (argc!=3){
```

```
  printf("Error....SINTAXIS: \n");
  printf("ejecutable 1 2 \n");
  printf("(1)  Archivo del texto en claro \n");
  printf("(2)  Archivo de errores \n");
  exit(1);
 }

/* Archivos de errores es un archivo nuevo */
 destino=fopen(argv[2],"w+");
 if(destino==NULL){
   printf("Error para crear archivo destino \n");
   exit(1);
 }

/* Abrir archivo origen */
  fuente=fopen(argv[1],"r");
  if(fuente==NULL){
   printf("Error en archivo destino \n");
   exit(1);
  }

/* Realiza las pruebas desde 1 hasta tope */
  tope=txtlen(argv[1],letras)/4;

  for(i=1;i<tope;i++){

   /* Define y validar tamanio de llave */
   llave=(int*)malloc(i*sizeof(int));
   if(llave==NULL){
    printf("Error en asignar tamanio de llave deseada \n");
    exit(1);
   }

  /* Crear archivo temporal */
   tempo=fopen("tempo.dat","w+");
   if(tempo == NULL){
    printf("Error en archivo temporal \n");
```

```
    exit(1);
   }

  /* Define el contenido de la llave aleatoriamente */
    fx(&llave[0],i,letras);

  /* Cifrar el archivo */
   rewind(fuente);
   k=0;
   while (!feof(fuente)){
    if (inversa((a=getc(fuente)),letras) < letras)
     putc(abc(cifrado(inversa(a,letras),((llave[k++%i])),letras),letras),tempo);
    else
     putc(a,tempo);
   }

  /* Cerrar archivo temporal */
   fclose(tempo);

  /* Libera espacio de memoria para llave utilizada */
   free(llave);

  /* Determinar el periodo */
    estimado=periodo("tempo.dat",letras);

  /* Comparar resultados: Periodo real contra el estimado */
   if(estimado != i)
    fprintf(destino,"%d\t%d\n",estimado,i);

  /* Realiza otra iteraci\'on */
  system("rm tempo.dat");
  }

 fclose(fuente);
 fclose(destino);
/* Fin de programa */ }
```

A.3 Evaluación de la determinación del corrimiento

Teniendo conocimiento del período, solo resta conocer el contenido de la llave involucrada, para lo cual en el Capítulo 3 hemos explicado el mecanismo seguido, pero sabemos que es importante evaluar la propuesta dada.

Se consideraron archivos de 500 caracteres hasta 50000 con una variación de 500 caracteres entre cada uno de los archivos. Cada uno de los archivos fueron cifrados con períodos que oscilaron entre $D = 1$ hasta un tercio del número de caracteres del texto cifrado.

Los pasos a considerar para evaluar el criptoanálisis del cifrado de Vigenére que hemos incluido fueron los siguientes:

1. *Cifrado de un texto*: Al archivo dado se le aplicó el cifrado de Vigenére mediante una llave cuyo contenido fué seleccionado aleatoriamente

2. *Estimar el periodo*: Se estima el período mediante el procedimiento definido en la sección 3.3.2.

3. *Estimar el corrimiento*: Se estima el el valor de la llave K mediante los pasos definidos en la sección 3.3.3.

4. *Comparar valores*: Se comparan los valores estimados con los reales. De existir errores ya sea en la estimación del período o en el contenido de la llave se almacena dicho evento.

Para obtener una refereńcia del código que realiza los pasos que hemos señalado, aquí lo incluimos:

```
/* Programa que cifra un archivo dado (argv[1]
   con una llave de 1 hasta MaxKey. definida
   como un tercio de la longitud del texto.
```

```
   Se cifra el archivo con una llave aleatoria
   se estima el periodo y el corrimiento utilizado
    para cada uno de los elementos de la llave,
    se compara valores, y si hay errores
    se almacen en (argv[2])
*/

#include "alib/criptoanalisis.h"
#define letras 26

void main (int argc, char *argv[]){
 int i=0, j=0, MaxKey, *llave, *llave1,res,suma=0;
 FILE *origen, *destino, *Error ;
 char letra;

/* Verifica el numero de argumentos */
if (argc!=3){
  printf("Error en el numero de argumentos \n");
  printf(" ejecutable (1) (2) \n");
  printf(" (1) Archivo en texto claro \n");
  printf(" (2) Archivo de errores \n");
  exit(1);
 }

/* Definir el tamanio de llave maximo a utilizar */
  MaxKey=txtlen(argv[1],letras)/3;

/* Abrir archivo de lectura y errores  */
 origen=fopen(argv[1],"r");
 Error=fopen(argv[2],"w+");

/* Validar entrada de datos */
 if(origen==NULL ||  Error==NULL){
  printf("Error en alguno de los archivos\n");
  exit(1);
 }
```

```
/* Realizar el cifrado desde 1 hasta MaxKey */
  for(i=1;i<MaxKey;i++){

    /* Abrir archivo destino, donde se cifrara la iteracion i-esima */
     destino=fopen("Tempo.dat","w+");

    /* Validar archivo utilizado */
     if(destino==NULL){
       printf("Error en archivo destino (main de Integral.c)\n");
       exit(1);
     }

    /* Reserva espacio para una llave de periodo i-esimo */
     llave=(int*)malloc(i*sizeof(int));

    /* Verifica que exista memoria suficiente */
     if(llave==NULL){
      printf("Tamanio de llave muy grande (Main Integral.c)\n");
      exit(1);
     }

    /* Define el contenido de la llave aleatoriamente */
     fx(&llave[0],i,letras);

    /* Realiza el cifrado, y almacena el resultado */
     j=0;
     rewind(origen);
     while (!feof(origen)){
       letra=getc(origen);
      if (inversa(letra, letras) >= letras)
        putc(letra,destino);
      else
        putc(abc(cifrado(inversa(letra, letras),llave[j++%i],letras), le
     }

/* Cierra el archivo cifrado */
```

```
    fclose(destino);

    res=periodo("Tempo.dat",letras);
    llave1=(int*)malloc(res*sizeof(int));


   /* Verifica que exista memoria suficiente */
    if(llave1==NULL){
     printf("Tamanio de llave muy grande (Main Integral.c)\n");
     exit(1);
    }

    for(j=0;j<res;j++)
     llave1[j]=0;

    corrimiento("Tempo.dat",letras,&llave1[0],res);

   /* Verifica resultado */
      if(res!=i){
fprintf(Error,"Periodo %d not \t %d\n",i, res);
      }else{
       suma=0;
       for(j=0;j<res;j++)
        suma+=llave[j]+llave1[j];
       if(suma!=0){
        fprintf(Error,"\nError en contenido de llave!! \n");
        for(j=0;j<res;j++)
         fprintf(Error,"%d not %d  \n ", llave1[j], llave[j]);
        fprintf(Error,"Fin de llave\n");
       }
      }

   /* Realiza otra iteracion */
   free (llave);
   fclose(destino);
   system("rm Tempo.dat");
  }
```

```
/* Fin de programa principal */
 fclose(origen);

/* Fin de archivo de Errores */
 fprintf(Error,"Fin de Archivo con Periodo maximo de= %d \n", MaxKey);
 fclose(Error);

}
```

A.4 Bibliotecas utilizadas

De los resultados obtenidos en la presente investigación hicimos uso de un sinnúmero de programas. Para facilitar el desarrollo utilizamos diversas funciones que cubrieran alguna tarea.

Incorporamos en pimer lugar la descripción de cada uno de las funciones y después incorporamos el código asociado.

criptoanalisis.h

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>
#include "fx.c"
#include "abc.c"
#include "cifrado.c"
#include "inversa.c"
#include "txtlen.c"
#include "periodo.c"
#include "index.c"
#include "fy.c"
#include "ordena.c"
#include "corrimiento.c"

/* Define el contenido de la llave aleatoriamente */
 void fx(int*,int, int);

/* Convierte un entero a letra  */
 char abc(int, int);

/* Cifra mediante suma modular */
 int cifrado(int,int,int);

/* Convierte una letra a su correspondiente entero */
 int inversa (char,int);
```

```
/* Determina el numero de caracteres, recibe el nombre del archivo */
 int txtlen(char*, int);

/* Determina el periodo utilizado, y el promedio de los IC */
 int periodo(char*, int);

/* Regresa el IC de un archivo dado, recibe el nombre de \'el */
 float index(char*, int);

/* Define la llave para cifrar/descifrar */
 void fy(int*,int,int);

/* Un archivo lo pone en columnas de 5 en 5, solamente caracters
   de la A-Z, omite enteros, y signos de puntuacion */
 void ordena(char*, char*, int);

/* Determina el contenido de una llave desconocida
   a partir del periodo estimado */
 void corrimiento(char*,int, int*, int);
```

abc.c

```
/* UN ENTERO LO ASOCIA A UN CARACTER

   Si se desea utilizar algun
   otro alfabeto, este se debe
   de incorporar en este
   archivo, asi como en el
   archivo "inversa.c"
*/

char abc(int numeral, int tope){
  char respuesta;

 if (tope == 27){
 switch(numeral){
```

```
  case 1: respuesta='B'; break;
  case 2: respuesta='C'; break;
  case 3: respuesta='D'; break;
  case 4: respuesta='E'; break;
  case 5: respuesta='F'; break;
  case 6: respuesta='G'; break;
  case 7: respuesta='H'; break;
  case 8: respuesta='I'; break;
  case 9: respuesta='J'; break;
  case 10: respuesta='K'; break;
  case 11: respuesta='L'; break;
  case 12: respuesta='M'; break;
  case 13: respuesta='N'; break;
  case 14: respuesta=''; break;
  case 15: respuesta='O'; break;
  case 16: respuesta='P'; break;
  case 17: respuesta='Q'; break;
  case 18: respuesta='R'; break;
  case 19: respuesta='S'; break;
  case 20: respuesta='T'; break;
  case 21: respuesta='U'; break;
  case 22: respuesta='V'; break;
  case 23: respuesta='W'; break;
  case 24: respuesta='X'; break;
  case 25: respuesta='Y'; break;
  case 26: respuesta='Z'; break;
  case 0: respuesta='A'; break;
  default : respuesta=' '; break;
 }
}

 if(tope == 26){
 switch(numeral){
  case 1: respuesta='B'; break;
  case 2: respuesta='C'; break;
  case 3: respuesta='D'; break;
  case 4: respuesta='E'; break;
```

```
   case 5: respuesta='F'; break;
   case 6: respuesta='G'; break;
   case 7: respuesta='H'; break;
   case 8: respuesta='I'; break;
   case 9: respuesta='J'; break;
   case 10: respuesta='K'; break;
   case 11: respuesta='L'; break;
   case 12: respuesta='M'; break;
   case 13: respuesta='N'; break;
   case 14: respuesta='O'; break;
   case 15: respuesta='P'; break;
   case 16: respuesta='Q'; break;
   case 17: respuesta='R'; break;
   case 18: respuesta='S'; break;
   case 19: respuesta='T'; break;
   case 20: respuesta='U'; break;
   case 21: respuesta='V'; break;
   case 22: respuesta='W'; break;
   case 23: respuesta='X'; break;
   case 24: respuesta='Y'; break;
   case 25: respuesta='Z'; break;
   case 0: respuesta='A'; break;
   default : respuesta=' '; break;
  }
 }

 return respuesta;
}
```

cifrado.c

```
/* CIFRADO UTILIZANDO SUMA MODULO M */

int cifrado (int vocal, int secreto, int letras){
  return (letras + vocal + secreto)%letras;
 }
```

fx.c

```
/* DEFINE EL CONTENIDO DE UNA LLAVE ALEATORIAMENTE */

 void fx(int *key1, int tope, int letras ){
  int j;
  srand(time(NULL));

  for(j=0;j<tope;j++)
   key1[j]=(rand()%letras);
 }
```

fy.c

```
/* DETERMINAR CONTENIDO DE LLAVE POR EL USUARIO */

void fy(int *llave, int periodo, int tope){
 int i;
 char datos[periodo];

 printf("Introduzca la llave ....\n");
 scanf("\n");
 gets(datos);

 for(i=0;i<periodo;i++)
  llave[i]= inversa(datos[i],tope);
}
```

inversa.c

```
/* CONVIERTE UNA LETRA A UN ENTERO */

int inversa (char letra, int tope){
  int respuesta;

 if (tope == 27){
  switch(letra){
  case 'A': case 'a': respuesta=0; break;
  case 'B': case 'b': respuesta=1; break;
  case 'C': case 'c': respuesta=2; break;
  case 'D': case 'd': respuesta=3; break;
  case 'E': case 'e': respuesta=4; break;
  case 'F': case 'f': respuesta=5; break;
  case 'G': case 'g': respuesta=6; break;
  case 'H': case 'h': respuesta=7; break;
  case 'I': case 'i': respuesta=8; break;
  case 'J': case 'j': respuesta=9; break;
  case 'K': case 'k': respuesta=10; break;
  case 'L': case 'l': respuesta=11; break;
  case 'M': case 'm': respuesta=12; break;
  case 'N': case 'n': respuesta=13; break;
  case '': case '': respuesta=14;  break;
  case 'O': case 'o': respuesta=15; break;
  case 'P': case 'p': respuesta=16; break;
  case 'Q': case 'q': respuesta=17; break;
  case 'R': case 'r': respuesta=18; break;
  case 'S': case 's': respuesta=19; break;
  case 'T': case 't': respuesta=20; break;
  case 'U': case 'u': respuesta=21; break;
  case 'V': case 'v': respuesta=22; break;
  case 'W': case 'w': respuesta=23; break;
  case 'X': case 'x': respuesta=24; break;
  case 'Y': case 'y': respuesta=25; break;
  case 'Z': case 'z': respuesta=26; break;
```

```
/* Definir casos especiales */
  case '' : case '' : respuesta=0; break;
  case '' : case '' : respuesta=4; break;
  case '' : case '' : respuesta=8; break;
  case '' : case '' : respuesta=15; break;
  case '' : case '' : respuesta=21; break;

  default : respuesta=100; break;
 }
}

if (tope == 26) {
  switch(letra){
  case 'A': case 'a': respuesta=0; break;
  case 'B': case 'b': respuesta=1; break;
  case 'C': case 'c': respuesta=2; break;
  case 'D': case 'd': respuesta=3; break;
  case 'E': case 'e': respuesta=4; break;
  case 'F': case 'f': respuesta=5; break;
  case 'G': case 'g': respuesta=6; break;
  case 'H': case 'h': respuesta=7; break;
  case 'I': case 'i': respuesta=8; break;
  case 'J': case 'j': respuesta=9; break;
  case 'K': case 'k': respuesta=10; break;
  case 'L': case 'l': respuesta=11; break;
  case 'M': case 'm': respuesta=12; break;
  case 'N': case 'n': respuesta=13; break;
  case 'O': case 'o': respuesta=14; break;
  case 'P': case 'p': respuesta=15; break;
  case 'Q': case 'q': respuesta=16; break;
  case 'R': case 'r': respuesta=17; break;
  case 'S': case 's': respuesta=18; break;
  case 'T': case 't': respuesta=19; break;
  case 'U': case 'u': respuesta=20; break;
  case 'V': case 'v': respuesta=21; break;
  case 'W': case 'w': respuesta=22; break;
  case 'X': case 'x': respuesta=23; break;
```

```
   case 'Y': case 'y': respuesta=24; break;
   case 'Z': case 'z': respuesta=25; break;

 /* Definir casos especiales */
   case '' : case '' : respuesta=0; break;
   case '' : case '' : respuesta=4; break;
   case '' : case '' : respuesta=8; break;
   case '' : case '' : respuesta=14; break;
   case '' : case '' : respuesta=21; break;

   default : respuesta=100; break;
  }
 }

  return respuesta;
 }
```

ordena.c

```
/* ORDENA UN ARCHIVO EN COLUMNAS DE 5 CON 5 ELEMENTOS */

void ordena (char *doc1, char *doc2, int tope){
 FILE *fuente, *destino;
 char a;
 int i=0;

/* Abrir archivos origen y destino */
 fuente=fopen(doc1,"r+");
 destino=fopen(doc2,"w+");

/* Validar estado de archivos */
 if(fuente==NULL){
  printf("Error en archivo origen (ordena.c)\n");
  exit(1);
 }
```

```
 if(destino==NULL){
  printf("Error en archivo origen (ordena.c)\n");
  exit(1);
 }

 while (!feof(fuente)){
  a=getc(fuente);
  if (inversa(a,tope) < tope){
   putc(abc(inversa(a,tope),tope), destino);
   i++;
   if(i%5 == 0)
     fprintf(destino,"    ");
   if(i%25 == 0)
     fprintf(destino,"\n");
   }
 }

 fclose(fuente);
 fclose(destino);
}
```

txtlen.c

```
/* CUENTA EL NUMERO DE CARACTERES VALIDOS */

int txtlen(char *archivo, int tope){
  int i=0;
  char letra;
  FILE *doc;

 /* Abrir archivo de conteo */
  doc=fopen(archivo,"r");

 /* Validar archivo utilizado */
  if(doc==NULL){
   printf("Error en archivo (txtlen.c)\n");
   exit(1);
```

```
  }

 /* Cuenta los caracteres */
  while(!feof(doc)){
    letra=getc(doc);
    if(inversa(letra, tope) < tope)
     i++;
   }

 /* Cierra archivo origen*/
 fclose(doc);
 return i;
}
```