
# Introducción a la criptografía

## *Métodos y protocolos criptográficos.*

**Organización de Datos (75.06)**
**Cátedra: Saubidet – Vera – Argerich**
**Facultad de Ingeniería. Universidad de Buenos Aires**

**Fecha de edición: Agosto 2004**

# *Índice*

Índice ..... 2
Introducción ..... 3
Conceptos Básicos de criptografía. ..... 4
Criptografía simétrica o de clave privada. ..... 7
Escitalo ..... 8
Caesar ..... 9
Sustitución Afín ..... 9
Cifrado Monoalfabético General ..... 10
Método PlayFair ..... 11
Criptosistema 'Hill' ..... 13
Cifrado de Vigènere ..... 15
Encriptación con clave privada por bloques ..... 16
DES y Triple DES ..... 18
Otros métodos de cifrado por bloques. ..... 20
Concejos a la hora de utilizar métodos criptográficos de clave privada. ..... 20
Criptografía asimétrica o de clave pública. ..... 22
KNAPSACKS. ..... 23
RSA. ..... 26
PGP : Pretty Good Privacy. ..... 28
Firmas digitales ..... 30
Protocolos en comunicaciones criptográficas ..... 32
Comunicaciones con criptosistemas simétricos ..... 33
Comunicaciones con criptosistemas asimétricos ..... 33
Comunicaciones con criptosistemas híbridos ..... 34
Firma digital con criptosistemas simétricos ..... 34
Firma digital con criptosistemas asimétricos ..... 35
Firma digital con message digest ..... 35
Comunicaciones con criptosistemas asimétricos y firma. ..... 36
Bit-Commitment ..... 36
Fair Coin Flips ..... 37
Zero-Knowledge Proofs ..... 37
Digital Cash ..... 38
Algunas anécdotas Criptográficas ..... 40
Bibliografía. ..... 44
Índice Alfabético ..... 45

## *Introducción*

Uno de los puntos que siempre estuvo en discusión sobre el almacenamiento de información en computadoras digitales y su distribución fue la seguridad de los mismos frente a posibles miradas indiscretas. Desde que la primera computadora hizo su aparición en alguna organización militar o gubernamental la necesidad de resguardar la información allí almacenada se hizo evidente.

Históricamente la criptografía y su ciencia paralela el criptoanálisis son disciplinas que experimentan marcados avances en épocas de guerra, allí es necesaria la comunicación de estrategias, planes tácticos e informes supersecretos entre las distintas fuerzas de cada bando de forma tal que si el código es interceptado por un eventual enemigo este no pueda hacerse de la información que se envío. La criptografía experimenta su mayor avance durante el transcurso de la segunda guerra mundial en donde adquiere un protagonismo singular, allí, el criptosistema de la maquina alemana 'Enigma' es roto por los criptoanalistas de las fuerzas aliadas siendo éste un factor que contribuyo en gran medida a la victoria final de los aliados.

La criptografía experimenta su segundo gran auge con la masificación de las comunicaciones digitales y el advenimiento de la era de las computadoras. La criptografía deja de ser un factor netamente militar y político para pasar a ser también un factor económico. Hoy en día suele ser necesario enviar y recibir mensajes a través de internet de forma tal que el mensaje solo pueda ser entendido por alguna persona en particular, por ejemplo si enviamos nuestro numero de tarjeta de crédito queremos que lo obtenga solamente el vendedor y no algún criptoanalista que pasaba por allí.

Generalmente se observa un juicio de valor donde se califica a los criptoanalistas como los 'malos' quienes intentan de robarnos información para hacernos daño. Sin embargo esto depende de la situación y de quien cuenta la historia pues en el caso de la segunda guerra mundial los criptógrafos alemanes eran los 'malos' y los criptoanalistas aliados eran 'los buenos', pero si nosotros queremos enviar un mensaje secreto a alguien y el mismo resulta publicado en un diario vamos a pensar que los criptoanalistas son personas sumamente viles. Como vemos todo depende de la situación.

En los últimos años la criptografía se ha convertido en una ciencia de enorme importancia y a la cual se le destina cada vez un tratamiento más serio y más científico de forma tal de lograr comunicaciones seguras. Como veremos esto no es una tarea para nada sencilla.

En el mundo de las computadoras la criptografía puede ser utilizada para varias cosas, algunas áreas importantes en donde se utiliza la criptografía son:

- La encriptación de información 'critica' que debe ser almacenada en computadoras, actos gubernamentales, informaciones secretas, etc.
- La encriptación de mensajes enviados a través de redes, redes locales, redes públicas e internet.
- La certificación de identidad de quienes envían mensajes importantes a través de internet.
- Protección de información 'delicada' que deba enviarse a través de internet como, por ejemplo, números de tarjetas de crédito.

- Encriptación de comunicaciones telefónicas, radiales o televisivas que pueden ser interceptadas.

## *Conceptos Básicos de criptografía.*

Etimológicamente la palabra Criptografía proviene del griego:

*Cripto* → oculto
*Grafía* → escritura

Según el diccionario de la real academia española se define como:

*"Arte de escribir con clave secreta o de un modo enigmático"*

Aunque para definirlo en forma completa se debe recurrir a una enunciación mas completa del término.

**Criptografía:**
*Ciencia que estudia la transformación de un mensaje en código, de forma tal que solo algunas personas puedan obtener el mensaje original a partir de dicho código*

Se puede ver una discrepancia entre la definición formal propuesta y la de la real academia. Esto es entendible puesto que la criptografía fue por siglos un arte hasta el año 1948 donde Claude Shannon enuncia la teoría de la criptografía moderna.

Existen dos trabajos fundamentales sobre los que se apoya prácticamente toda la teoría criptográfica actual. Uno de ellos, desarrollado por Shannon – nombrado anteriormente - en sus artículos "A Mathematical Theory of Communication" (1948) y "Communication Theory of Secrecy Systems" (1949) donde sienta las bases de la Teoría de la Información y de la Criptografía moderna. El segundo, publicado por Whitfield Diffie y Martin Hellman en 1976, se titulaba "New directions in Cryptography", e introducía el concepto de Criptografía de Llave Pública, abriendo enormemente el abanico de aplicación de esta disciplina.

Entre las disciplinas que engloba la criptografía cabe destacar la teoría de algoritmos, teoría de la información y matemática discreta.

Aunque a primera vista el único propósito de la criptografía es mantener en secreto un mensaje (confidencialidad) a la vista de ojos no autorizados (y efectivamente este fue su primer uso), también se utiliza la criptografía para otros fines:

- Autenticación: Consiste en la posibilidad que debe tener la persona que recibió el mensaje de asegurarse que quien lo envió realmente es quien dice ser.
- Integridad: Consiste en capacidad que debe tener quien recibió el mensaje que el mismo no fue alterado total o parcialmente en el camino.
- No repudio: Consiste en la imposibilidad que debe existir que quien envió el mensaje falsamente niegue su autoría.

Por lo tanto:

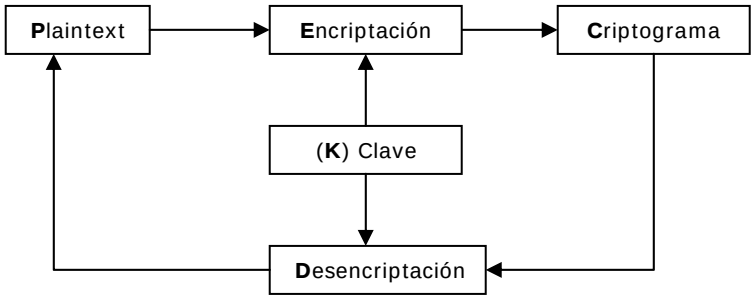
**Objetivos de la criptografía**
La criptografía debe proveer confidencialidad, autenticación, integridad y no repudio

Para entender cabalmente la ciencia de la criptografía y los diferentes métodos existentes es necesario manejar una terminología común. En primer lugar se puede ver que en todos los métodos se pueden identificar 3 conjuntos bien definidos:

- El conjunto de los mensajes: Corresponde a la totalidad de los mensajes sin cifrar que pueden ser tomados por el método para su posterior encriptación. Recibe el nombre de **texto plano** o **plaintext (P)**.
- El conjunto de mensajes cifrados: Corresponde a la totalidad de mensajes cifrados que pueden obtenerse luego del proceso de encriptación. Recibe el nombre de **criptograma (C)**.
- El conjunto total de claves: Corresponde a la totalidad de valores que pueden ser utilizados en el método de encriptación y desencripción de los mensajes. Se conoce como **clave (K)**

Por otro lado esto seria inservible sin la existencia de 2 mecanismos algorítmicos, uno para la encriptación y otro para la desencriptacion.
Se puede ver el proceso de encriptación y desencriptacion como cajas negras donde ante la entrada de un mensaje y una clave se obtiene como salida un nuevo mensaje encriptado o desencriptado según sea el proceso invocado.

Gráficamente:




Cabe aclarar que no necesariamente se utiliza la misma clave para el proceso de encriptación y desencriptacion. Conociéndose a los que utilizan las mismas claves como **criptografía simétrica o privada** y a su contraparte criptografía **asimétrica o pública**.

**Criptosistema**:
Se define un criptosistema como una terna (M,C,K), donde:
- M representa el conjunto de todos los mensajes sin cifrar (plaintext) que pueden ser procesados.
- C representa el conjunto de todos los posibles mensajes cifrados, o criptogramas.
- K representa el conjunto de claves que se pueden emplear en el criptosistema.
Mas un método de encriptación y otro de desencriptación.

La necesidad de de encriptar un mensaje surge ante el problema de transmitir un mensaje a una o mas personas por un medio inseguro donde existen personas con la intención de interceptar el mismo. Tenemos registros en la historia de métodos de encriptación utilizados por los griegos (conocido como Escitalo) y por los romanos

(Metodo Caesar) con fines militares. Y seguramente tan vieja como estos métodos, existieron personas que trataron de burlarlo para conocer el contenido de los mensajes enviados. Estas personas dieron origen a otra ciencia.

**Criptoanálisis:**
Ciencia que intenta 'romper' los criptosistemas desarrollados por los criptógrafos, para obtener el mensaje a partir del código cifrado

Aunque existen diversas formas de romper un criptosistema, que incluye la obtención de forma *non-sancta* de la clave, el criptoanálisis se entiende de una forma bastante relajada: Encontrar una debilidad en el criptosistema que pueda ser explotada con una complejidad menor a la fuerza bruta. No importa si un ataque de fuerza bruta requiera $2^{128}$ pruebas, un ataque que requiera $2^{110}$ pruebas será considerado un quiebre. El ataque puede requerir grandes e irreales cantidades de texto plano escogido o una necesidad de almacenamiento descomunal, simplemente el encontrar una evidencia que el criptosistema no se comporta como fue publicitado se considera un quiebre.

De todas formas el objetivo último del criptoanálisis es descifrar un mensaje sin conocer la llave o bien obtener a partir de uno o más criptogramas la clave que ha sido empleada en su codificación. Todo aporte en ese sentido por lo tanto es bienvenido. Criptoanálisis exitosos de variantes reducidas de criptosistemas le pueden permitir a otros criptoanalistas a eventualmente (incluso años después) extender el quiebre a las versiones completas.

No se considera criptoanálisis el descubrimiento de un algoritmo secreto de cifrado. Un método que su fuerza se recuesta en su secreto esta condenado tarde o temprano al desastre. El algoritmo de encriptación tarde o temprano es descubierto. Es por eso que muchas veces quienes inventan un criptosistema publican su resultado e inician concursos o recompensas a quienes pueden vulnerarlos. Un ejemplo de eso ocurrió con el primer criptosistema de clave publica: "Knapsack" (más detalles en la sección de algunas anécdotas criptográficas).

Obviamente el objetivo de todo criptógrafo es lograr que el método utilizado no pueda ser vulnerado nunca o en un tiempo razonable. Se verá inclinado a la primera opción, pero la existencia de un método **incondicionalmente seguro** – como es conocido – es difícil de lograr, en tanto que los métodos **Computacionalmente seguros** son corrientes y fáciles de implementar. Entran a la clasificación de computacionalmente seguros aquellos que demandan un tiempo y recursos excesivamente elevado de procesamiento mediante una computadora o varias de ellas para lograr romper la encriptación.

La definición de Computacionalmente seguro es relativa a la tecnología. Por lo tanto el avance de la misma (mayor velocidad de procesamiento principalmente) hace que ciertos criptosistemas que alguna vez pertenecieron a esta categoría dejen de serlo. A esta categoría pertenece el algoritmo DES propuesto por la NSA (National Security Agency) de EEUU. Este criptosistema fue utilizado hasta que se demostró públicamente que existía la tecnología para descifrar cualquier criptograma en corto tiempo. Más detalles sobre el episodio del Método DES en la sección "Algunas anécdotas criptográficas".

Por ultimo queda una interesante pregunta: Existe el código indescifrable?

Según palabras de Edgar Allan Poe, escritor norteamericano aficionado a la criptografía:

*"es dudoso que el género humano logre crear un enigma que el mismo ingenio humano no resuelva"*

La respuesta a esta pregunta sin embargo es positiva. En 1917, J. Mauborgne y G. Vernam inventaron un criptosistema perfecto. Dicho sistema consistía en emplear una secuencia aleatoria de igual longitud que el mensaje, que se usaría una única vez —lo que se conoce en inglés como one-time pad—, combinándola mediante alguna función simple y reversible con el texto claro carácter a carácter.

Al ser una clave aleatoria y de longitud igual al mensaje, no hay redundancias, cualquier criptograma puede corresponder a cualquier mensaje.

Este método presenta el grave inconveniente de que la clave es tan larga como el propio mensaje, y si disponemos de un canal seguro para enviar la clave, ¿por qué no emplearlo para transmitir el mensaje directamente?

Otro inconveniente es la creación de claves aleatorias. Actualmente es posible crear tiras de bits pseudoaleatorias por ejemplo grabando con un micrófono el ruido ambiente.

No obstante los problemas planteados este método fue puesto en práctica utilizando libros con tiras de caracteres como clave que a medida que se iban utilizando eran destruidos. Tanto el emisor como el receptor tenían una copia de ese libro por lo tanto el problema se reducía a lograr que el libro les llegue a ambos (sin que un tercero en el camino obtenga una copia).

## *Criptografía simétrica o de clave privada.*

Se puede caracterizar a la criptografía de clave privada a aquella que utiliza tanto para encriptar como para desencriptar una misma clave.

Se puede diferenciar dos casos:

- Mensaje encriptado que es guardado por el creador del mismo él (y solo él) conoce la clave.
- Mensaje encriptado para enviar a una o varias personas, entonces la clave tiene que ser conocida tanto por el emisor como los receptores.

En el primer caso la utilización de estos métodos es ideal. Se tiene una sola clave y por lo tanto esta es más fácil de recordar. En el segundo caso la clave tiene que ser distribuida a todos los participantes de la comunicación. Ahí radica la debilidad de estos criptosistemas, enviar la clave es un serio problema, pues esta puede caer en manos equivocadas en el intermedio.

Estos sistemas fueron y son ampliamente utilizados. Y dada la amplia variedad de criptosistemas existentes se presenta una clasificación que permite separarlos de acuerdo a sus métodos de encriptación.

En Primer un criptosistema de clave privada puede utilizar la **transposición** o la **sustitución**. Donde la transposición consiste en el intercambio de los caracteres del mensaje de su lugar y la sustitución consiste en – como lo indica su nombre – sustituir los caracteres de un mensaje por otros.

Por otro lado los métodos de sustitución se pueden clasificar según sean:

- **Monoalfabéticos**: Cuando se encripta, cada caracter encriptado corresponde a un caracter del mensaje original y viceversa (Ejemplo “R” es “23”).

- **Homofónicos**: Cuando un caracter de texto original se encripta en varios caracteres del texto encriptado (por ejemplo “A” puede ser “16” o “47”).

- **Poligráficos**: Cuando n caracteres del mensaje se encriptan juntos originando caracteres del mensaje encriptado (Por ejemplo “OHA” es “RSW” y “OHB” es “QWD”).

- **Polialfabéticos**: En los cifrados polialfabéticos la sustitución aplicada a cada carácter varía en función de la posición que ocupe éste dentro del texto claro. En realidad corresponde a la aplicación cíclica de n cifrados monoalfabéticos. De esta forma se atenúan las propiedades estadísticas del mensaje

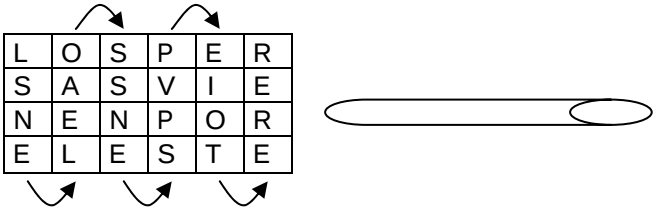
A continuación se darán ejemplos de algunos criptogramas de cada uno de los distintos tipos nombrados.

## Escitalo

Tipo: Transposición

El escitalo, utilizado por los griegos, estaba formado por un bastón cilíndrico con un radio particular y una tira de piel que se enrollaba alrededor de aquél. El texto se escribía a lo largo del bastón y sólo podría ser leído si se disponía de otro bastón de dimensiones similares. Al leer la tira desenrollada el mensaje no tiene sentido.

Por ejemplo si quiero escribir: “LOSPERSASVIENENPORELESTE” (Se quitaron los espacios)
Enrollo la tira y escribo:




Desenrollado se lee:

“LSNELEAOSSNESPVPEIOTERER”

Si bien este método es impracticable con una computadora (no puedo enrollar el monitor en un palo… :-p ) su originalidad y simplicidad permite generar métodos similares utilizando la computadora por ejemplo realizando sustituciones de un carácter con otro a “n” posiciones del mensaje.

**Criptoanálisis:**

Los sistemas de Transposición pura no se utilizan sino en conjunto con otros métodos. Su principal debilidad es que mantienen estadísticamente las mismas apariciones de las letras en el mensaje cifrado del texto plano. El idioma tiene ciertas letras que se suelen utilizar mas seguido asimismo también se observan grupo de letras que suelen aparecer seguidos uno del otro (“DE”, “EL”, “QU” por ejemplo). Es cuestión de buscar estas letras y grupos y ver que distancias guardan unas de las otras para empezar a probar posibles transposiciones.

## Caesar

Tipo: Sustitución Monoalfabético.

El criptosistema Caesar le debe su nombre al emperador romano de ese mismo nombre. Según la leyenda Caesar invento este método como medio seguro de para comunicar ordenes a sus generales. Consiste en rotar en un valor de 1 cada letra de un mensaje a transmitir. Con este mecanismo la “A” → “B”, la “B” → “C”, “C” → “D”, … “Z” → “A”.

Por ejemplo si el Ceaser quería enviar el siguiente mensaje: “VENGAAROMA”.
Escribía:
“WFOHBBSPNB”

El general al recibir el mensaje debía restarle 1 al mensaje recibido y leer lo que el Ceaser quiso decir.

Este mecanismo en si es muy simple. Se agrega complejidad determinando que la sustitución en vez de ser de 1 lugar sea de un valor “N”, donde ahora la clave es “N”.

**Criptoanálisis:**

Para el sistema 'CAESAR' la tarea de un criptoanalista es realmente sencilla, pues la cantidad de posibles claves de este sistema es muy limitada. Trabajando con un alfabeto de 25 caracteres hay solamente 25 posibles claves (1..25) la clave 26, es idéntica a la clave 1, la clave 27 es idéntica a la 2 y así sucesivamente. De esta forma el criptoanalista puede chequear una por una las 25 posibles claves y observando el resultado obtenido se llega fácilmente y en muy poco tiempo al mensaje original.

Este es un criptosistema cuyo punto débil es el espacio de claves, como hay muy pocas claves posibles la técnica mas recomendable para el criptoanalista es simplemente probar todas las posibles claves. A este método se lo denomina 'ataque por fuerza bruta' y cuando el tiempo estimado para el ataque es razonable es un método infalible.

## Sustitución Afín

Tipo: Sustitución Monoalfabético.

La sustitución afín se puede considerar una variante del método Caesar, que agranda el universo de las claves a dos valores “A” y “B”.

Suponiendo que trabajamos con un alfabeto de N=26 caracteres (tomaremos como ejemplo las letras del alfabeto ingles), las letras se numeran en orden alfabético de forma tal que A=0, B=1,..., Z=25
Todas las operaciones aritméticas se realizan en la forma modulo N es decir que aN=0, aN+1=1, aN+2=2 etc. (“a” entero mayor o igual a cero)

La transformación de cada carácter seria de acuerdo a la formula:

$$E(a,b)(M) = (aM + b) \bmod N$$

Siendo: a<N, b<N, a y N relativamente primos.

**Definición:**
Dos números 'a' y 'b' son relativamente primos si su Máximo común divisor es 1. El concepto de números relativamente primos es de gran importancia en el mundo de la criptografía, este y otros conceptos de la teoría de números son utilizados muy a menudo para encriptar y desencriptar información.

El requerimiento de que a y b sean relativamente primos asegura que la función f(h) = aM + b sea inyectiva ya que si la función no fuera inyectiva un cierto caracter en el mensaje cifrado podría corresponder a mas de un caracter en el mensaje original y no podríamos descifrar el mensaje. Por ejemplo si a=10 y b=1 entonces tanto la 'a' como la 'n' son mapeados al caracter 'b'.
En nuestro ejemplo '3' y '26' son relativamente primos mientras que '10' y '26' no lo son (MCD=2).

Por ejemplo si a=3 y b=5 el texto 'hola' se encripta de la forma:

h=7 → E(h)= 7 * 3 + 5 = 26 mod 26 = 0 = a
o=14 → E(o)= 14 * 3 + 5 = 47 mod 26 = 21 = v
l=11 → E(l) = 11 * 3 + 5 = 38 mod 26 = 12 = m
a=0 → E(a)= 0 * 3 + 5 = 5 mod 26 = 5 = f

El mensaje 'hola' encriptado es 'avmf'

**Criptoanálisis.**

Un criptosistema afín también es vulnerable a un ataque por fuerza bruta, ya que existen 12 posibles valores para a (con un alfabeto de 26 simbolos): 1,3,5,7,9,15,17,19,21,23,25 y 26 posibles valores para b (0..25) de esta forma la cantidad de claves posibles es 12*26=312 pero se debe eliminar el caso (a=1,b=0) pues no encripta, luego la cantidad de posibles claves es 311 y el sistema es completamente vulnerable a un ataque por fuerza bruta.

## Cifrado Monoalfabético General

Tipo: Sustitución Monoalfabético.

El Cifrado Monoalfabético General es el caso más general de cifrado monoalfabético. La sustitución ahora es arbitraria, cualquier símbolo en el texto plano puede corresponder a cualquier símbolo del criptograma (Pero siendo un método monoalfabetico esa correspondencia se da siempre para el mismo par de símbolos utilizando la misma clave).
Se utiliza como método de codificación una tabla de sustitución, existiendo como universo de tablas posibles todas las combinaciones de correspondencia entre símbolos. Se considera la clave a un numero “K” que identifica al numero de tabla de sustitución a utilizar.
En el caso de utilizar un alfabeto de N símbolos tendremos N! tablas de sustitución posibles y por lo tanto N! claves.

**Criptoanálisis:**

El cifrado monoalfabético en si constituye la familia de métodos más simple de criptoanalizar, puesto que las propiedades *estadísticas* del texto claro se conservan en el criptograma. Emparejando las frecuencias relativas de aparición de cada símbolo en el mensaje cifrado con el histograma de frecuencias del idioma en el que se supone está el texto claro, podremos averiguar fácilmente la clave.

Se muestra a continuación algunas tablas con estadísticas de aparición de letras en el idioma ingles, francés y castellano.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| FRANCES | E | S | A | R | N | U | T | L | I | O | TOTAL |
| FRECUENCIA % | 15 | 8 | 6 | 5,5 | 5,4 | 4,8 | 4,7 | 4,6 | 4,5 | 4 | 54,5% |
| INGLES | E | T | A | O | I | N | S | H | R | | TOTAL |
| FRECUENCIA % | 10 | 8,2 | 7 | 6,5 | 6,4 | 6,3 | 6 | 4 | 3,6 | | 61,1% |

Cálculo de la frecuencia de las 10 letras más utilizadas en INGLÉS y FRANCÉS, procedentes de una muestra de 10.000 signos (letras y espacios), procedente de un artículo de Roberto Vacca y Ana Goñi en la revista Newton

| Altas | Medias | bajas | |
|---|---|---|---|
| E – 16,78% | R – 4,94% | Y – 1,54% | J – 0,30% |
| A – 11,96% | U – 4,80% | Q – 1,53% | Ñ – 0,29% |
| O – 8,69% | I – 4,15% | B – 0,92% | Z – 0,15% |
| L – 8,37% | T – 3,31% | H – 0,89% | X – 0,06% |
| S – 7,88% | C – 2,92% | G – 0,73% | K – 0,00% |
| N – 7,01% | P – 2,776% | F – 0,52% | W – 0,00% |
| D – 6,87% | M – 2,12% | V – 0,39% | |

Frecuencias relativas de las letras del idioma castellano. Estos datos proceden del libro 'Estudio lexicométrico del diario "El País"', de Enrique Fontanillo, en el que se toman como muestra los ejemplares de dicho diario publicados durante una semana (52619 letras en total).

## Método PlayFair

Tipo: Sustitución Poligráfico

El método **Playfair** Este sistema criptográfico debe su nombre al Baron Playfair de St Andrews.
Las letras del alfabeto omitiendo la 'J' son distribuidas en una matriz de 5x5 arbitrariamente.
El mensaje a encriptar es dividido en bloques de dos caracteres cada uno con la consideración de que ningún bloque debe estar formado por dos ocurrencias de la misma letra, si así fuera se intercala una letra o se cambia el mensaje original de forma que esto no ocurra.

Para cada par de dos caracteres si los mismos no se encuentran en la misma fila o columna se determina el rectángulo formado por los dos caracteres y se encripta tomando los caracteres que están en las esquinas del rectángulo y en la misma fila que el carácter a encriptar.
Si los dos caracteres se encuentran en la misma fila los encriptamos tomando el carácter que se encuentra a la derecha de los mismos
Si los dos caracteres están en la misma columna los encriptamos tomando el carácter que se encuentra debajo de los mismos.

En el sistema Playfair si bien no es cierto que todo caracter es siempre encriptado en un mismo caracter si vale que todo par de caracteres siempre es encriptado en el mismo par de caracteres, por lo que en lugar decimos que el sistema es poligráfico de orden 2.
Por ejemplo sea el siguiente mensaje: “VIVAELFAIRPLAY”
Y la siguiente matriz (que es también la clave):

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

La salida seria: “XFWKZMHKBDCESG”

Veremos algunos pasos de encriptación para entender como funciona:

Para el mensaje anterior tomamos los primeros 2 caracteres: “VI”, luego formamos el rectángulo en la matriz teniendo en cuenta que los caracteres no se encuentran en la misma fila o columna:

| | |
|---|---|
| **I** | F |
| Z | N |
| U | K |
| D | O |
| X | **V** |

Y el código resultante es el formado por los vértices que quedan dentro del rectángulo que no son los caracteres del texto plano pasaran a ser parte del criptograma. La “V” pasa a ser la “X” (se encuentra en la misma fila que el carácter “V”) y la “I” pasa a ser la “F”

Por otro lado si consideramos los caracteres “EL”, vemos que estos se encuentran en la misma fila, por lo tanto:

| | | | | |
|---|---|---|---|---|
| **L** | M | **E** | Z | N |

En este caso “EL” pasa a ser “ZM”. De haber estado la “E” En la posición de la “N” y por lo tanto al borde derecho de la matriz el carácter a reemplazar seria el primer elemento de la matriz en esa misma fila (En este caso la “L”)

Para el tercer caso supongamos que tenemos que encriptar los caracteres “GM” y en la matriz estos se encuentran en la misma columna:


Este caso se resuelve análogamente al caso de la fila pero tomando el carácter que se encuentra debajo de cada carácter. Por lo tanto “GM” pasa a ser “WA”

**Criptoanálisis:**

El sistema Playfair es un sistema de encriptación bastante bueno, la cantidad de posibles claves es enorme ya que son las permutaciones de 25 elementos tomados de entre 26 lo cual da un número muy grande como para derrotar al algoritmo por fuerza bruta. Además es un sistema polígráficos por lo que un análisis de la frecuencia de aparición de cada carácter en el código cifrado no nos aporta nada.

La técnica que se debe utilizar con el esquema 'Playfair' consiste en analizar la frecuencia de aparición de los pares de letras (diagramas) y compararlas con los diagramas mas frecuentes del idioma en el cual se supone que se escribió el mensaje original, en castellano los diagramas más probables son (Ordenados por frecuencia):

ES,EN,EL,DE,LA,OS,AR,UE,RA,RE,ER,AS,ON,ST,AD,AL,OR,TA,CO

El criptoanalista deberá analizar cual es el diagrama mas ocurrente en el código cifrado y ver que ocurre si se lo reemplaza por 'ES', de esta forma se van probando distintas combinaciones entre los diagramas mas frecuentes en el mensaje cifrado y los diagramas mas frecuentes del idioma hasta que se consigue descifrar el texto. Esta es una técnica muy habitual del criptoanálisis y suele funcionar muy bien.

Los sistemas polígráficos de orden 'N' son vulnerables a este tipo de ataque, en general todo criptoanalista esta preparado con programas específicos para analizar las frecuencias de aparición de caracteres individuales, diagramas, tetragramas, y compararlas con las frecuencias estadísticas de un determinado idioma, los programas se encargan también de generar las pruebas necesarias y todo lo que tiene que hacer el criptoanalista es analizar si el texto que resulta de las pruebas tiene sentido. Incluso esto puede hacerlo el programa comparando el texto contra un diccionario.

## Criptosistema 'Hill'

Tipo: Sustitución Polígráfico

El Criptosistema 'Hill' esta basado en el álgebra lineal y ha sido importante en la historia de la criptografía.
Suponiendo que trabajamos con un alfabeto de 26 caracteres. Las letras se numeran en orden alfabético de forma tal que A=0,B=1,Z=25
Todas las operaciones aritméticas se realizan en la forma modulo 26 es decir que 26=0, 27=1, 28=2 etc.
Elegimos un entero d que determina la dimensión de las matrices a utilizar, las matrices serán de dxd elementos.
Los elementos de la matriz de dxd serán enteros entre 0 y 25, además la matriz M debe ser inversible usando aritmética modulo 26.

Por ejemplo:

$$M = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}, Mi = \begin{pmatrix} 15 & 17 \\ 20 & 9 \end{pmatrix}$$

3*15 + 3*20 = 45 + 60 = 45 mod 26 + 60 mod 26 = 19 + 8 = 27 (27 mod 26 = 1)
2 * 17 + 5 * 9 = 79 (79 mod 26) = 1
etc...

Si se hacen todas las cuentas modulo 26 observamos que M*Mi nos da la matriz identidad, luego Mi es la inversa de M.

Dado un mensaje a encriptar debemos tomar bloques del mensaje de "d" caracteres y aplicar:

MP=C  C es el código cifrado para el mensaje P.

Ejemplo:

Sea

$$M = \begin{pmatrix} 3 & 3 \\ 2 & 5 \end{pmatrix}$$

Si queremos encriptar el mensaje 'HELP' debemos encriptar los cuatro caracteres de 'HELP' en bloques de 2 caracteres cada uno, el primer bloque

$$P1 = 'HE' = \begin{pmatrix} 7 \\ 4 \end{pmatrix}$$

$$P2 = 'LP' = \begin{pmatrix} 11 \\ 15 \end{pmatrix}$$

$$M * P1 = \begin{pmatrix} 7 \\ 8 \end{pmatrix}$$

$$M * P2 = \begin{pmatrix} 0 \\ 19 \end{pmatrix}$$

Luego 'HELP' encriptado equivale a 'HIAT'.

Este sistema es poligráfico de orden “d” pues para cada conjunto de “d” caracteres la salida encriptada es la misma. Sin embargo dos apariciones de un conjunto de caracteres menores a “d” no necesariamente dan el mismo resultado.

Para desencriptar el método es idéntico al anterior pero usando como es evidente la matriz inversa de la usada para encriptar.

**Criptoanálisis**

Como se puede ver el sistema de Hill plantea a los criptoanalistas problemas mucho mayores a los que planteaba 'CAESAR'. Para empezar el espacio de claves es mucho mayor, en este caso es de 4C25, es decir las permutaciones de 4 elementos tomados de entre 25 posibles. Y usando una matriz más grande la cantidad de posibles claves se puede hacer tan grande como sea necesario para hacer que sea imposible un ataque por fuerza bruta.

Lo mejor que puede hacer un criptoanalista es tratar de conseguir un código para el cual se conozca una parte del mensaje. Y ver si con ambos datos es capaz de encontrar cual fue la matriz utilizada para encriptar el mensaje.

## Cifrado de Vigènere

Tipo: Polialfabetico.

Es un ejemplo típico de cifrado polialfabético que debe su nombre a Blaise de Vigènere, su creador, y que data del siglo XVI. La clave está constituida por una secuencia de símbolos

K = {k0, k1, . . . kd-1}, y que emplea la siguiente función de cifrado:

Ek(mi) = [mi + k(i mod d)] mod n

Siendo: d la cantidad de símbolos de la clave.
n la cantidad de símbolos en el alfabeto utilizado.
mi el i-ésimo símbolo del texto claro.
k(i mod d) el (i mod d)-ésimo simbolo de la clave

Por ejemplo:

Si consideramos al alfabeto de entrado como los números del 0 al 9 (n=10)
K = {“4”,”5”,”8”} d =3
Y el mensaje: “479014”

Entonces:
Ek(4) = (4 + 4) mod 10 → 8
Ek(7) = (7 + 5) mod 10 → 2
Ek(9) = (9 + 8) mod 10 → 7
Ek(0) = (0 + 4) mod 10 → 4
Ek(1) = (1 + 5) mod 10 → 6

Ek(4) = (4 + 8) mod 10 → 2

Salida: “827462”

**Criptoanálisis:**

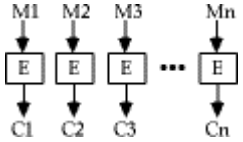
Para criptoanalizar este tipo de claves basta con efectuar d análisis estadísticos independientes agrupando los símbolos según la ki empleada para codificarlos. Para estimar d, buscaremos la periodicidad de los patrones comunes que puedan aparecer en el texto cifrado. Obviamente, para el criptoanálisis, necesitaremos al menos d veces más cantidad de texto que con los métodos monoalfabéticos.

## *Encriptación con clave privada por bloques*

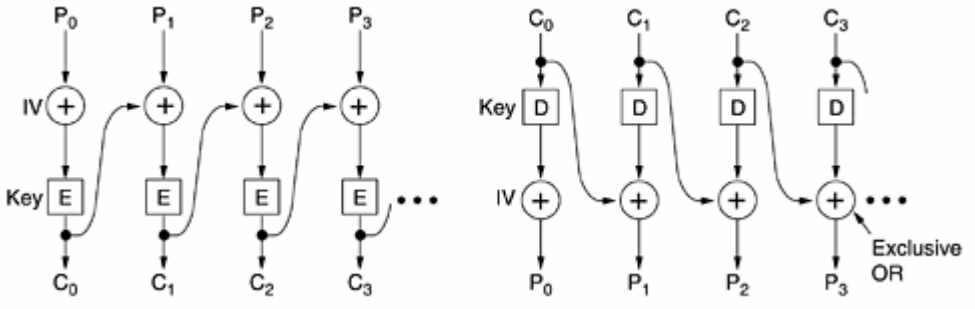
Los métodos explicados en el apartado anterior se conocen como métodos de encriptación por Stream (chorro), en ellos la encriptación es aplicada bit a bit. En esta sección analizaremos otros tipos de criptosistemas de clave privada y son los conocidos como los criptosistemas por bloques.
En la encriptación por bloques se procede a dividir el texto plano en bloques de un tamaño determinado para ser luego procesados.

Existen diferentes formas de procesar los bloques entre ellos:

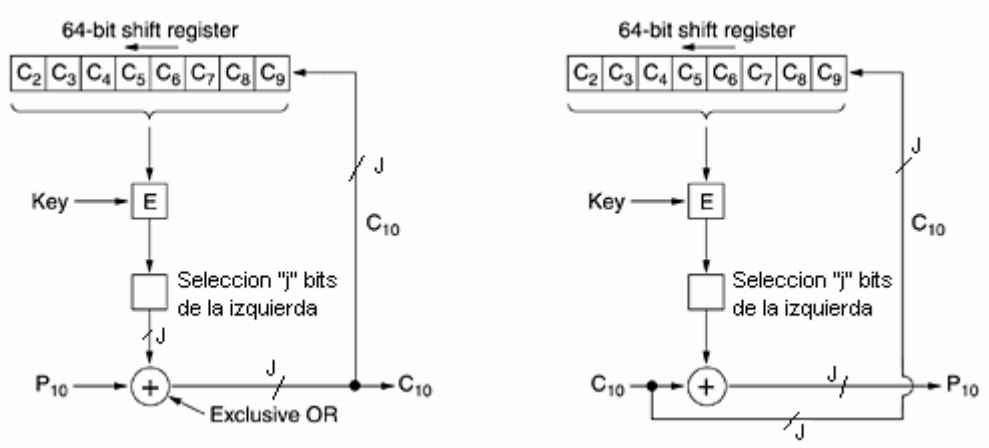
- **ECB (Electronic Code Block):** Este el método mas simple, consiste en dividir el bloque a comprimir en porciones de igual longitud que la clave y encriptar cada uno en forma independiente.



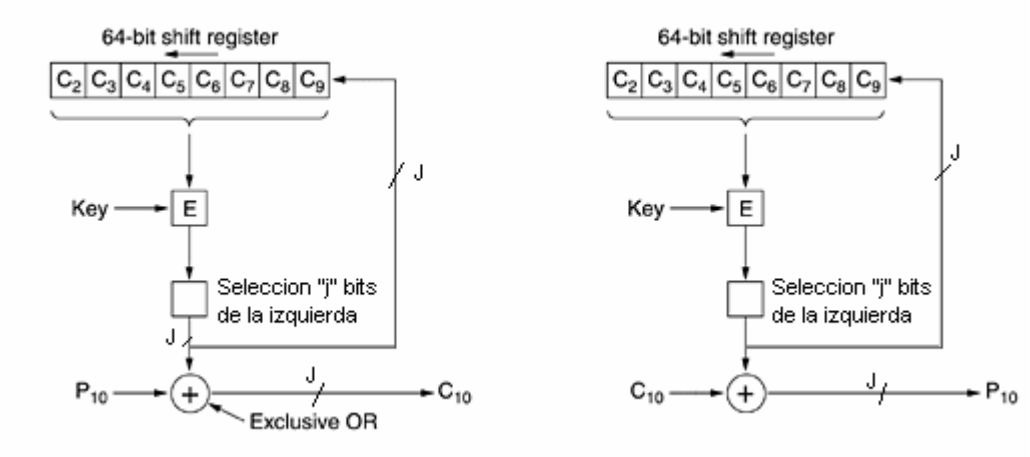

- **CBC (Cipher Block Chainning):** En este método para encriptar cada bloque se utiliza la salida del bloque anterior. Básicamente a la entrada de cada bloque (el texto plano) se le realiza la operación de XOR con la salida del bloque anterior y luego es encriptado. De esta forma se logra que bloques con entrada idéntica de texto plano produzcan diferentes criptogramas. Por otro lado si se produce algún error en un bit en el proceso de desencriptacion este es arrastrado en todos los procesos siguientes. En el bloque inicial se utiliza un bloque dummy (Conocido como IV: initialization vector), este no necesariamente debe ser secreto pero no debe ser reutilizado con la misma clave en mas de una oportunidad.




- **CFB (Cipher Feedback):** Este método permite utilizar bloques de tamaño de "j" bits y encriptarlos. Para realizar este proceso comienza con un bloque IV al que le aplica la encriptación con la clave y luego toma los "j" bits de la izquierda del resultado y les realiza un XOR con los elementos del bloque del texto plano obteniendo con eso el criptograma. Para los próximos "j" bits del texto plano se realiza nuevamente el mismo proceso pero utilizando como bloque de entrada a encriptar al IV realizando un shifteo a la izquierda de "j" posiciones y ingresando en los "j" bits de la derecha el criptograma calculado en la ronda anterior.




- **OFB (Output Feedback):** Básicamente es similar al CFB aunque en vez de tomar al criptograma resultante como parte de la entrada del proceso de encriptación siguiente lo que utiliza son los "j" bits de la salida luego de encriptar antes de realizar el XOR. La ventaja de esto radica en la no propagación de errores antes un bit incorrecto en el criptograma.



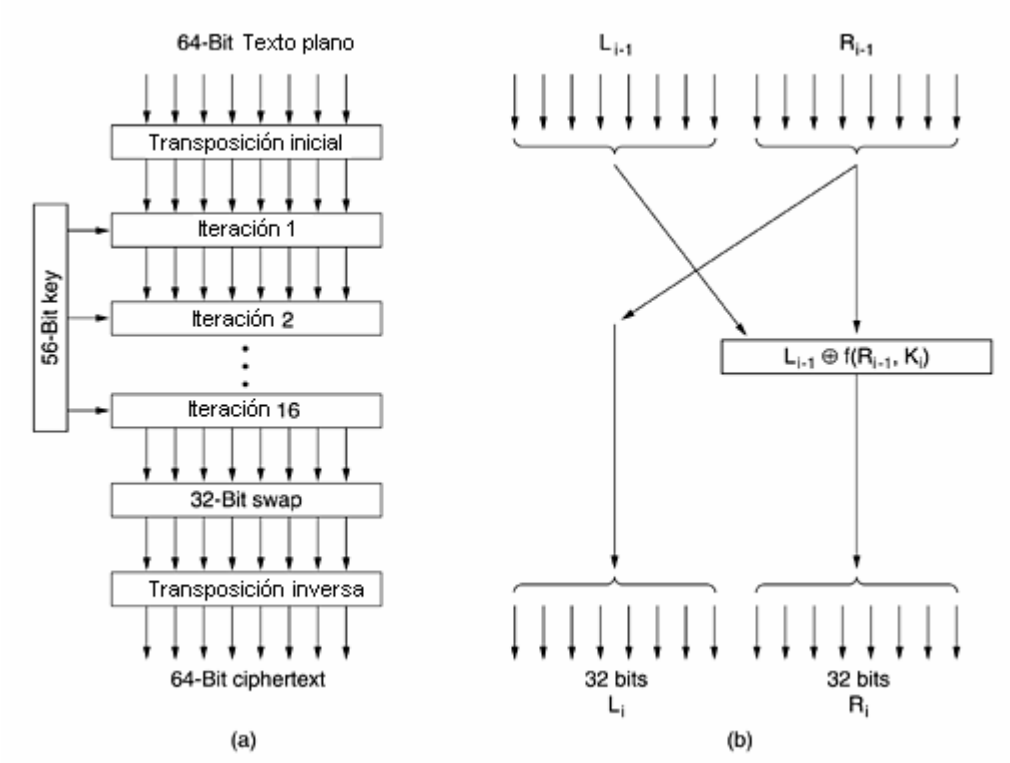

Sobre la forma en que trabaja cada método en más detalle puede consultarse el standard ANSI/FIPS correspondiente.

Otra cuestión a tener en cuenta en el cifrado por bloques es que puede ocurrir que al tener un tamaño fijo el último bloque debido a la longitud del texto plano no alcance a llenarse. Se debe buscar un método de completar este bloque y algunos de estos métodos es utilizar algún carácter que no aparezca en el texto plano u otros medios de tal forma que pueda reconstruir el mensaje los caracteres agregados puedan ser eliminados sin problemas.

Para la encriptación se utilizan diferentes tamaños de claves desde 56, 64, 128 a 1024 bits de tamaño. El tamaño del bloque depende en gran medida del tamaño de la clave. Las funciones de encriptación dependen del método y estos se pueden representar mediante la operación de sucesivas sustituciones y transposiciones entre otras operaciones básicas.

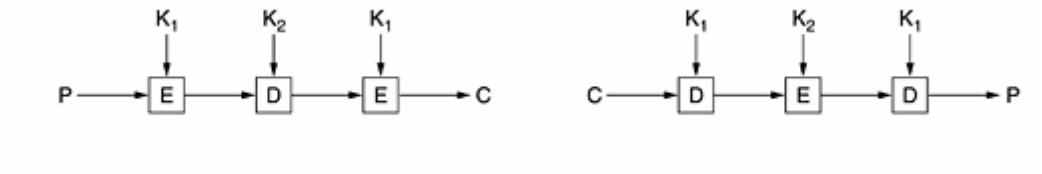
## DES y Triple DES

El método DES (Data Encryption Standard) fue desarrollado por IBM y estandarizado por el NBS (Nacional Bureau of Standards) y estandarizado en 1976 como estandar Federal Information Processing Standard FIPS.
Utiliza claves de 56 bits en bloques de 64 bits y permite la utilización de los 4 métodos de procesamiento de bloques explicados anteriormente.




Para realizar la encriptación utiliza 16 rondas (iteraciones) donde procesa el texto plano (a). Cada iteración se comporta como una red de Feistel (b) donde la entrada es dividida en dos partes y la parte izquierda es procesada mediante un XOR al resultado de la parte Derecha luego de aplicarle una función junto con la clave convirtiéndose el resultado final en la parte derecha de la salida. Mientras que la parte izquierda de la salida es la parte derecha de la entrada.
En cada iteración se le aplica una transformación a la clave para que esta cambie de un proceso al otro. La función utilizada en la red de Feistel son cuatro pasos que involucran sustituciones y transposiciones entre otras operaciones.

En el año 1993 se demostró que el método dejo de ser seguro y por lo tanto se busco un reemplazante (Para mas información leer la sección “Algunas anécdotas Criptográficas” bajo el nombre “DES(cifrado)”). Temporalmente se utilizo una variante del DES conocida como “Triple DES” que es no mas ni menos que el mismo método en encriptación pero utilizando 2 claves diferentes de 56bits y construyendo el criptograma realizando en primer lugar una encriptación con DES con la clave 1, desencriptando el resultado con la segunda clave y por ultimo encriptando nuevamente el resultado con la primer clave:




La desencriptación se realiza invirtiendo el proceso con las mismas claves.

## Otros métodos de cifrado por bloques.

Existen otros métodos de cifrado por bloques que se utilizaron o se siguen utilizando. Entre ellos:

- IDEA (Internacional Data Encryption Algorithm): Desarrollado por el Instituto de Tecnología Federal de Suiza para sustituir al DES en 1990. El cifrado se realiza por bloques de 64bits y claves de 128 bits. Es implementable tanto en software como hardware.

- SkipJack: Desarrollado por el U.S. Nacional Security Agency en 1980 para las telecomunicaciones. Utiliza bloques de 64bits y claves de 80bits. Realiza 32 rondas (a comparación de las 16 de DES)

- RC5: Desarrollado por Ron Rivest, en los años 1994-95. Permite cifrar en bloques de 32, 64 o 128bits y la clave puede variar de 0 hasta 2040bits. Ademas permite determinar la cantidad de rondas a realizar (de 0 a 255).

- AES (Advanced Encryption Standard): EL NIST (National Institute Of Standards Technology) convoco un concurso para el desarrollo de un sistema de encriptación simétrico para utilizarse durante los próximos 20 años. Este debía ser tan seguro y rápido como el triple DES y funcionar tanto como cifrador de bloques como de fluyo (entre otras características solicitadas). A finales del 2000 se selecciono al método Rijndael como el ganador entre 15 candidatos seleccionados en 1998). Algunos otros finalistas fueron Twofish, Serpent, RC6, y Mars.

## *Concejos a la hora de utilizar métodos criptográficos de clave privada.*

Los algoritmos de clave privada pueden construirse tan eficientes como se desee utilizando passwords más y más largos, sin embargo por más largo que sea el password estos algoritmos presentan una vulnerabilidad evidente: el password.

Aun teniendo el mejor sistema criptográfico si no tenemos cierto cuidado nos exponemos a exponer información a personas no autorizadas. Viéndolo con una analogía aun teniendo la mejor caja fuerte si dejamos la llave en el cerrojo no tendrán problema en abrirla. En nuestro caso la llave es la clave y en el momento de seleccionar una clave se tiene que tomar en cuenta ciertas reglas para aumentar nuestra seguridad:

1. Deben ser memorizadas. (Jamás escrita en un papel)
2. Suficientemente complejas.
3. Carecer de significado.
4. Fáciles de recordar.
5. Deben ser modificadas con frecuencia.

En un esquema de encriptación por clave privada todo aquel que conozca el password es capaz de desencriptar un mensaje, de aquí que a veces, es mas importante estudiar como proteger el password que como trabaja el algoritmo elegido. Además, muchas

veces es necesario transmitir, o enviar el password a alguna persona por lo que será necesario a su vez encriptar el password ingresando en un loop infinito.

Muchas veces el password es tan vulnerable que los criptoanalistas no se molestan en descifrar el código interceptado sino que directamente intentan averiguar el password. Uno de los ejemplos más habituales consiste en averiguar passwords que permiten el acceso a determinados sistemas: cuentas bancarias, computadoras, computadoras donde se guardan otros passwords, etc.
A la hora de distribuirla y mantenerla tenemos que tener en cuenta que existen diversos medios mediante los cuales un intruso puede intentar obtenerla o lograr ingresar a nuestros datos.

Entre ellas:

- Caminos inseguros: A veces la necesidad de enviar una clave de una persona a otra hace peligrar el secreto de la clave. La existencia de caminos pocos seguros o la intermediación de personas no confiables. La difusión de una clave privada es un asunto complicado de por si. La intercepción de la misma puede provocar enormes daños.

- Ataque por diccionario: Consiste en la utilización de un juego de claves como prueba (diccionario) altamente utilizadas o relacionadas con el dueño de la clave para probar ingresar. Si no se tienen en cuenta las reglas para la creación de clave se puede producir una intrusión no autorizada con este método. (por ejemplo usar como clave fechas de cumpleaños o nombre de mascotas)

- Fuerza Bruta: Consiste en la prueba de todas las combinaciones posibles de clave en un determinado criptosistema. En este caso si nuestra clave es corta será más fácil de romper. Aquí entra en juego tambien el universo de claves en el criptosistema elegido.

- Shoulder Surfing: Esta técnica es la más básica y consiste en merodear a aquellas personas que conocen el password que se quiere averiguar intentando ver si se consigue visualizar el momento en que el password es tipeado en un teclado o escrito en algún papel, variantes más modernas de esta técnica incluyen programas residentes que monitorean las teclas que se oprimen en el teclado, cámaras que registran lo que se tipea desde un punto elevado, etc. La forma mas elemental es como su nombre lo indica observar por encima del hombro de la persona que tipea el password, parece tonto pero se utiliza muchísimo.

- Caballos de Troya: Los caballos de Troya son programas que se diseñan con el fin específico de robar passwords. El programa es introducido en una computadora y lo que hace es simplemente cada vez que es ejecutado pedirle el password al usuario y si este lo tipea (grave error) guardarlo en un archivo. Luego lo único que hay que hacer es cada tanto consultar el archivo y ver que es lo que nuestro caballo de Troya ha 'pescado'.

- Ingeniería Social: Esta disciplina puede parecer ridícula pero es la más exitosa en cuanto a robo de passwords. La Ingeniería Social consiste en conseguir que una persona, simplemente, le diga su password a otra. Las técnicas son de lo mas variadas: llamados telefónicos pidiendo el password pues se cayó un disco y hay que backupear la información de cada usuario, pedidos de password para

'verificaciones rutinarias', encuestas a ver quien tiene el password mas seguro o original, etc, etc... Aunque parezca mentira hay personas realmente especializadas en este tipo de ataques.

La fuerza de un criptosistema de clave privada no reside únicamente en el algoritmo utilizado, o en la longitud de la clave sino que depende, también, del mensaje a enviar. Hay mensajes que por sus características propias serán mas fáciles de descifrar que otros, dado un mismo criptosistema.

Recomendaciones a la hora de escribir un texto altamente critico que deba ser encriptado por clave privada:

- No utilizar espacios en blanco, escribir todo el texto de corrido. Cualquier caracter de alta probabilidad de ocurrencia, como por ejemplo los espacios en blanco, son un punto débil en el mensaje aun cuando se utilice un esquema polialfabético.

- Si es muy necesario separar las palabras del mensaje a enviar, utilizar cualquier caracter elegido arbitrariamente en reemplazo del espacio en blanco.

- Escribir con mucho cuidado el texto intentando que la distribución de cada caracter utilizado en el texto sea lo mas pareja posible, esto es de gran importancia, evitar el uso de uno o mas caracteres en forma predominante, usar palabras sinónimos, o incluir faltas de ortografía y sintaxis si es necesario. Escribir por ejemplo 'salujdhos pedfdro' que puede ser entendido fácilmente cuando se descifra el código pero podría llegar a complicar sensiblemente el criptoanálisis.

- Empezar algunas palabras con caracteres que no tengan sentido alguno como por ejemplo la llave que cierra '}' o un punto y coma ';', esto puede desalentar varios intentos criptoanalíticos correctamente orientados.

- Escribir algunas palabras al revés o con todas las vocales al final y las consonantes al principio, una especie de doble codificación. Por ejemplo: 'sldsauo qrdueio pdreo' quiere decir 'saludos querido pedro'. Los criptoanalistas que consideren como factor cierto la alternancia de vocales y consonantes en un texto tendrán problemas para descifrar nuestro mensaje. En algunas palabras se pueden poner todas las vocales al final y en otras todas al principio.

- Jamas utilizar dos veces la misma palabra en un texto, aun en sistemas polialfabéticos esto constituye una debilidad.

- Escribir todos los mensajes de la forma mas breve que sea posible, evitar el uso de artículos salvo que sean estrictamente necesarios para comprender el texto. Las chances de éxito de un buen criptoanálisis aumentan vertiginosamente a medida que se consigue más información sobre el mensaje a descifrar.

## *Criptografía asimétrica o de clave pública.*

La aparición de este tipo de criptosistemas es reciente: 1976 en la publicación “New directions in Cryptography” de Whitfield Diffie y Martin Hellman. Se puede caracterizar

a la criptografía de clave pública a aquella que utiliza diferentes claves para encriptar y desencriptar.

Dentro del par de claves una recibe el nombre de clave privada y la otra clave publica. La clave privada es guardada en secreto por el dueño de la clave y la pública es otorgada a todo aquel que nos quiera enviar un mensaje encriptado.

Los mensajes que son encriptados con la clave pública de un usuario solo pueden ser desencriptados con la clave privada del mismo. Por otra parte en algunos métodos lo encriptado con la clave privada puede ser desencriptado con la clave pública (y esto es utilizado en firmas digitales entre otras aplicaciones).

El mecanismo de encriptación es el siguiente:

- La persona que nos quiere enviar un mensaje utiliza nuestra clave publica para encriptarlo y nos envía el mensaje
- Al recibir el mensaje los desencripta con la clave privada y se obtiene el mensaje original.

## KNAPSACKS.

El primer ejemplo sobre el cual presentaremos a los sistemas de clave publica esta basado en un problema de ingenio denominado 'el problema de la mochila'

**'El problema de la mochila'**

*Se tiene una mochila con capacidad para "K" kilos. Además se cuenta con una lista de "n" objetos cuyos pesos se conocen y son (A1,A2,A3,...,AN). El problema consiste en seleccionar una cierta cantidad de objetos de la lista de forma tal que la mochila quede completamente llena.*

Para resolver este problema no se conoce ningún método mejor que el ir probando las distintas combinaciones de objetos y ver si en alguna de ellas la mochila queda completamente llena. Si los pesos de los objetos están en un vector (A1,A2,A3,...,AN) una combinación puede escribirse como un numero binario de N bits en el cual un uno en la posición "i" indica que el elemento "i" debe estar en la mochila, un cero indica lo contrario.

Ej:

A=(3,45,6,7,21,12,9,90)

C1=(00000001)=90
C2=(10101010)=3+6+21+9=39
etc...

De esta forma se puede ver claramente que la cantidad de combinaciones a probar es 2^N siendo N el numero de elementos del vector, a este tipo de vectores lo denominaremos 'Knapsack'. Si N es un numero lo suficientemente grande (por ejemplo

300) la cantidad de combinaciones a probar es tan grande que resultaría imposible probarlas todas antes de que se termine el universo!.

Veamos como podemos construir un sistema de clave publica utilizando Knapsacks.

Empecemos por inventar un sistema de clave privada que utilice 'Knapsacks'

Dado un mensaje lo pasamos a binario y tomamos bloques de "N" bits de acuerdo al tamaño del Knapsack, sumamos los elementos del Knapsack que corresponden a los bits en 1 y el número resultante es la codificación del bloque.

Ej: Con bloques de 8 bits y usando el código Ascii si el vector es: (3,45,6,7,21,12,9,90)

'H' = 01001000 = 45 + 21 = 66
'o' = 01101111 = 45 + 6 + 21 + 12 + 9 + 90 = 183
'l' = 01101100 = 45 + 6 + 21 + 12 = 84
'a' = 01100001 = 45 + 6 + 90 = 141

'Hola' = 66,183,84,141

Hasta ahora tenemos un método para encriptar un mensaje, la clave privada por el momento es el Knapsack, sin embargo este sistema tiene un problema: el receptor 'legal' del mensaje pese a poseer el Knapsack aun tiene que resolver el problema de la mochila para poder descifrar el mensaje, esto obviamente dista mucho de ser lo deseado. La solución consiste en simplificar el problema de la mochila utilizando un vector super-incrementante (super increasing, tradúzcanlo mejor si pueden).

**Definición:**
Un vector super-incrementante es aquel en el cual el elemento Ak es mayor a la sumatoria de todos los elementos con subíndice menor que él.

Ej: (1,3,5,11,21,44,87,175,349,701) es super-incrementante.

Con un vector super-incrementante resolver el problema de la mochila es fácil, para un cierto numero lo que hay que hacer es tomar el primer elemento del vector menor o igual al numero y poner su bit en 1, luego al número restarle el elemento y con el resultado repetir el procedimiento hasta que el número se hace cero.

Supongamos que tenemos el numero 734.

Para 734: Como 734 > 701 => el bit numero 10 es 1

734-701 = 33

Para 33: 33 > 21 => el bit 5 es 1

33-21 = 12

Para 12 : 12 > 11 => el bit 4 es 1

12-11 = 1

Para 1 : El bit 1 es 1

Luego 734=(1001100001) = 1+11+21+701

El algoritmo de desencriptado, como puede verse, es ahora muy sencillo. El problema es que el sistema sigue siendo de clave privada, conociendo el Knapsack se puede tanto encriptar como desencriptar cualquier mensaje.

**Convirtiendo los Knapsacks en un sistema de clave pública**

Lo que debemos lograr es identificar cual va a ser la clave privada y cual va a ser la clave publica, para ello una vez que tenemos un vector super-incrementante lo que se hace es elegir dos números t y m forma tal que t y m no tengan factores en común.

Al numero t lo denominamos multiplicador.
Al numero m lo denominamos modulo.

Además debemos encontrar un numero t' que sea el inverso multiplicativo de t usando aritmética modulo m.

Ejemplo:

Sea

m=1590 Verificar que 1590 y 43 no tienen factores en común
t=43 (43 es primo y 1590 no es divisible por 43)
1590 y 43 son relativamente primos.

Para el inverso de t hacemos 1591/43 = 37 luego t'=37
(37 * 43 = 1591, 1591 mod 1590 = 1)

Ahora lo que hacemos es aplicarle a cada elemento Ai del Knapsack la función
Ai'= Ai*t mod m

(1,3,5,11,21,44,87,175,349,701)=>(43,129,215,473,903,302,561,1165,697,1523)

Al vector que obtenemos una vez aplicada la función lo utilizaremos como clave publica (notar que el vector obtenido no es super-incrementante). Es decir que para encriptar un mensaje utilizaremos el método que ya hemos descripto utilizando la clave publica. Si el mensaje cifrado es interceptado el interceptor tiene que resolver el problema de la mochila para poder desencriptar el mensaje, y como sabemos este problema le va a llevar mucho mas que toda su vida.

La clave privada esta formada por t' y m, el receptor legal del mensaje primero convierte todos los números haciendo c*t' mod m y luego convierte el Knapsack de la misma forma, lo único que le resta hacer es resolver el problema de la mochila con un vector super-incrementante, lo cual es fácil.

Ejemplo:

Para el Knapsack ejemplo anterior la clave publica es (tomando solo 8 elementos para no hacer tantas cuentas, obviamente cuanto mayor es el vector mas seguro es el sistema)

P = (43,129,215,473,903,302,561,1165)

Supongamos que queremos encriptar 'Hola'

'H' = 01001000 = 129 + 903 = 1032
'o' = 01101111 = 129 + 215 + 903 + 302 + 561 + 1165 = 3275
'l' = 01101100 = 129 + 215 + 903 + 302 = 1549
'a' = 01100001 = 129 + 215 + 1165 = 1509

'Hola' = 1032,3275,1549,1509

Si este código es interceptado y sabiendo la clave publica no puede hacerse nada para desencriptar el mensaje pues la clave publica no es super-incrementante (aquí reside la gracia del asunto).

La clave privada es t'=37 m=1590

Para desencriptar usamos la clave privada:

1032 * 37 mod 1590 = 24
3275 * 37 mod 1590 = 335
1549 * 37 mod 1590 = 73
1509 * 37 mod 1590 = 183

A continuación se convierte el vector publico en un vector super-incrementante utilizando la misma transformación.

Luego con los números (24,335,73,183) y el vector super-incrementante se puede desencriptar fácilmente el mensaje.

## RSA.

El algoritmo de clave publica más probado y utilizado en todo el mundo es el algoritmo RSA, denominado así debido a sus autores: Rivest, Shamir y Adleman.

Está basado en una idea asombrosamente sencilla de la teoría de números y hasta la fecha ha resistido todo tipo de ataques criptoanalíticos.

La idea es simple: dados dos números primos p y q muy grandes es sencillo a partir de p y q hallar su producto (p*q) pero es un problema muy complejo a partir del producto hallar los números p y q en cuestión. Si bien hasta el momento no se ha podido demostrar que la factorización prima de un número es un problema NP-complejo, todos los intentos realizados para factorizar un número en forma veloz han fracasado.

Sean dos números p y q primos de aproximadamente 100 dígitos cada uno.

$$n = p*q \quad y \quad \S(n) = (p-1) * (q-1)$$

Además se elige un número random d de muchos dígitos tal que d y §(n) son relativamente primos. Y un número e, 1<e<§(n) tal que e*d=1 usando aritmética módulo §(n).

n = módulo.
e = exponente de encriptación.
d = exponente de desencriptación.

La clave pública estará formada por n y e.
La clave privada estará formada por p,q,§(n) y d.

Para encriptar se pasa el mensaje a binario y se lo divide en bloques de un cierto tamaño, cada bloque se encripta elevando el numero a la potencia e y reduciéndolo modulo n. Para desencriptar se eleva el código a la potencia d y se lo reduce modulo n.

$$E(M) = M^e \bmod n = C$$
$$D(C) = C^d \bmod n = M$$

El tamaño de los bloques es i tal que $10^{(i--1)} < n < 10^i$
Ejemplo (chiquito para poder seguir las cuentas):

Sea p=5 , q=11, n=p*q=55,§(n)=40.
Elegimos d=23 pues 23 y 40 son relativamente primos.
Luego e=7 pues 7*23=161 (161 mod 40) = 1.

Si encriptamos números comprendidos en el rango (0..15) (tenemos 4 bits)

| Número | Encriptado | |
|---|---|---|
| 0 | 0 | (0^ 7 mod 55) |
| 1 | 1 | (1^ 7 mod 55) |
| 2 | 18 | (2^ 7 mod 55) |
| 3 | 42 | (3^ 7 mod 55) |
| 4 | 49 | (4^ 7 mod 55) |
| 5 | 25 | (5^ 7 mod 55) |
| 6 | 41 | (6^ 7 mod 55) |
| 7 | 28 | (7^ 7 mod 55) |
| 8 | 2 | (8^ 7 mod 55) |
| 9 | 4 | (9^ 7 mod 55) |
| 10 | 10 | (10^ 7 mod 55) |
| 11 | 11 | (11^ 7 mod 55) |
| 12 | 23 | (12^ 7 mod 55) |
| 13 | 7 | (13^ 7 mod 55) |
| 14 | 9 | (14^ 7 mod 55) |
| 15 | 5 | (15^ 7 mod 55) |

Probar que el desencriptado funciona correctamente, por ejemplo para desencriptar el 42 debemos hacer 42^23, esta operación puede hacerse fácilmente sin usar números 'super enormes' ya que por cada producto aplicamos un modulo n.

42^2=4 42^4=16 42^8=36 42^16=31 42^17=37 42^18=14 42^19=38
42^20=1 42^21=42 42^22=4 42^23=3

Luego 3 mod 55 = 3 y queda desencriptado.

Notar que para calcular las potencias trabajamos siempre con aritmética modulo n.

El ejemplo presentado tiene algunas falencias que pueden ser descubiertas fácilmente por el lector (lo dejamos como ejercicio), estas fallas se reducen automáticamente a valores casi nulos cuando los números p y q son lo suficientemente grandes.

Una peculiaridad mas del metodo RSA es que si desencripto un texto plano con mi clave privada y el receptor lo encripta con la clave publica se vuelve a obtener el texto plano:

$$E(D(M)) = M$$
$$D(M) <> M$$

Esto hecho permite utilizar el método para las firmas digitales como veremos mas adelante.

**Criptoanálisis:**

Las técnicas criptoanalíticas más utilizadas contra el RSA, aunque sin éxito, consisten en intentar factorizar el numero "n" que se distribuye en la clave publica averiguando de esta forma los números p y q. Debido a que no existen algoritmos eficientes para factorizar un número, el problema de descomponer un numero muy grande insume un tiempo tan elevado que los ataques mas sofisticados contra el RSA han fallado (o casi...)

El algoritmo RSA sin embargo presenta una vulnerabilidad: hay una leyenda que indicaría que el algoritmo es vulnerable.
Hasta el momento el ataque mas devastador contra el RSA ocurrió en el siglo XVII y hasta el día de la fecha no se ha vuelto a repetir. Se supone que el misterioso Fermat se ha llevado a la tumba un método para factorizar números que hasta el día de hoy nadie ha descubierto. Mientras no aparezca otro Fermat el RSA permanece tranquilo. Para mas detalles leer “Factorizando primos...” en la sección de algunas anécdotas criptográficas.

## PGP : Pretty Good Privacy.

En esta sección analizamos el programa más utilizado para encriptar y desencriptar datos mediante algoritmos de clave publica. El PGP se utiliza en internet y en casi todas las redes de mensajería cada vez que quiere transmitirse información privada.

PGP es un producto de distribución libre, es distribuido con sus fuentes y su distribución le ha causado a su autor Philip Zimmerman más de un problema como veremos más adelante.

PGP trabaja con el algoritmo RSA utilizando claves de 256,512 o 1024 bytes según el nivel de seguridad que se necesite, las claves de 1024 bytes superan ampliamente los más estrictos requisitos militares sobre seguridad criptográfica.

PGP genera las claves públicas y privadas del usuario utilizando un algoritmo muy avanzado de pseudoaleatorización que mide los tiempos transcurridos entre lo que se tipea en un teclado. (PGP solicita al usuario que tipee durante un cierto tiempo en la pantalla) o los movimientos del mouse (se solicita al usuario que lo mueva aleatoriamente durante cierto tiempo).

La clave publica queda grabada en el disco y lista para ser distribuida, la clave privada se almacena también en el disco, PGP en sus manuales destaca que el acceso a la computadora donde se almacena la clave privada debe restringirse en forma drástica pues el conseguir la clave privada anula todo el sistema, el autor recomienda el uso de dispositivos que distorsionen las señales de radio en el ambiente donde reside la computadora pues existen dispositivos ultra-avanzados de las agencias gubernamentales que permiten leer la información de un disco a distancia mediante ondas de radio (!!).

Las claves públicas que nos envían otros usuarios son almacenadas en un conjunto de claves públicas (Public-key-ring) sobre el cual se pueden realizar altas, bajas y modificaciones.

Cuando un usuario le envía a otro su clave pública, por ejemplo a través de internet, el usuario que recibe la clave suele querer chequear que la clave publica recibida sea la del usuario que el quiere y no cualquier otra. Para ello PGP permite extraer de cada clave publica un numero conocido como 'FINGERPRINT' el Fingerprint puede ser chequeado telefónicamente o personalmente, y si coincide puede certificarse que la clave publica es de quien dice ser. (Cualquier cambio en la clave pública modifica el Fingerprint). El fingerprint se calcula hasheando la clave pública.

PGP dispone de varias opciones interesantes:

- Envío de mensajes en forma clásica:

  Este esquema sigue el mecanismo clásico de la encriptación por clave pública, el mensaje es encriptado usando la clave publica de un determinado usuario de forma tal que solo pueda ser desencriptado por la clave privada del mismo.

- Certificación de mensajes:

  Esta es una utilidad muy recomendable, y sirve para que un usuario firme un mensaje de forma tal que se pueda autenticar su autoría. Lo que hace el PGP es primero extraer un 'concentrado' del mensaje sometiéndolo a una función de hashing, luego el concentrado es encriptado con la clave privada del usuario y agregado al final del mensaje. Cuando el mensaje es recibido por un usuario la firma digital es desencriptada usando la clave pública del usuario y luego el mensaje es sometido a la función de hashing, si el concentrado coincide con el concentrado desencriptado del mensaje entonces el mensaje fue escrito por quien dice ser, de lo contrario o bien fue escrito por otra persona o bien fue modificado el texto del mensaje.

Los mensajes certificados a su vez pueden ser encriptados para que solo puedan ser leídos por una cierta persona.

Notar que la certificación utiliza el juego de claves en forma inversa al uso normal de las mismas.

- Mensajes solo para sus ojos:

  Esta opción del PGP permite encriptar un mensaje para una cierta persona de forma tal que cuando esta lo desencripte usando su clave privada el texto del mensaje solo se pueda ver en pantalla y no pueda ser grabado en un archivo, esta opción otorga una seguridad extra a quien envía el mensaje y tiene miedo que el usuario que lo recibe lo trate en forma descuidada dejándolo por allí.

- Borrado especial del archivo a encriptar:

  Cuando se quiere encriptar un mensaje muy critico que esta escrito en un archivo, PGP dispone de la opción de eliminar el archivo original del disco una vez encriptado. PGP no utiliza un borrado común del archivo sino que sobreescribe el área del disco con sucesivas pasadas de unos, ceros y unos y ceros alternados en forma random, esto lo hace varias veces. El algoritmo de borrado del PGP asegura que la información no podrá ser recuperada del disco. (Si el algoritmo no es lo suficientemente seguro el análisis de trazas magnéticas del disco puede permitir recuperar la información).

PGP es un programa sumamente seguro y es utilizado en todo el mundo para el envío de e-mail en forma segura y la certificación de mensajes de importancia.

Las leyes federales del gobierno de Estados Unidos prohiben la exportación de sistemas criptográficos que utilicen claves mayores a 40 bits, como PGP las utiliza y el programa fue puesto en el dominio publico su autor Philip Zimmerman fue sometido a juicio por el gobierno de los Estados Unidos. Durante el juicio los manuales que venían con el PGP solicitaban ayuda económica para el autor quien estaba en la cárcel (no se cobraba ni un peso por el software sino que se pedía una colaboración voluntaria)

A fines de 1996 el juicio fue declarado nulo.

## *Firmas digitales*

Las firmas hechas a mano son utilizadas como medio de prueba de autoría o acuerdo de un documento desde hace ya largo tiempo.

La firma tiene un gran poder sobre el documento y los firmantes. En primer lugar convence al receptor que el firmante deliberadamente rubrica el documento y deja un claro testimonio que el firmante y no otro es quien lo realiza. Además impide el repudio por parte del firmante de su acuerdo en el documento.

Por otra parte la firma es no reusable, pertenece al documento y no puede ser utilizada para validar otro. Y el documento una vez firmado es inalterable.

Estas premisas con respecto a las firmas hicieron muy difícil su aplicación en el campo informático donde copiar u alterar un documento es extremadamente fácil, así como no dejar traza del hecho. No obstante con algo de ingenio se puede aplicar firmas digitales para autenticar un documento.

La firma digital es capaz de lograrse mediante criptosistemas simetricos, no obstante esto es del todo satisfactorio por que se requiere de una tercera persona en la comunicación que funcione como intermediario y haga de certificador y por lo tanto determine que el documento enviado no ha sido modificado y fue entregado. Sin embargo es la criptografía de clave asimétrica la que permitió la utilización de la fima digital en una forma sencilla y segura.

La firma digital con clave asimétrica no requiere de intermediarios y es por si misma prueba de quien envió el mensaje. La "firma" del mensaje consiste en desencriptar el texto plano a enviar con la clave privada propia. Para comprobar la firma el receptor ni bien recibe el mensaje debe encriptar con la clave pública de quien envía el mensaje lo recibido y obtendrá el texto plano. De esta forma puede verificar que el emisario del mensaje es quien dice ser (pues la única forma de leer el mensaje es utilizando la clave publica del emisor para volver inteligible el mensaje recibido) y que el mensaje no ha sido alterado (si un tercero intenta modificarla no contara con la clave privada del emisor por lo tanto no conseguirá crear un mensaje que al aplicar la transformación con la clave publica del emisor reconstruya el texto plano).

Se requiere por lo tanto que:

$$E(D(M)) = M \qquad \text{Además del habitual } D(E(M)) = M$$

No todos los criptosistemas asimétricos cumplen con la propiedad solicitada. RSA es uno que si cumple por lo tanto es utilizado para firmas digitales.

El método anterior asegura autenticación y secreto (tener en cuenta que para leer el mensaje hay que quitar la firma encriptando el mensaje con la clave publica del emisor). En ciertas circunstancias se quiere que el mensaje pueda ser leído independientemente de la firma. Por lo tanto se propuso un mecanismo alternativo para la firma de documentos. Este mecanismo utiliza lo que se conoce como message digest.

**Definición:**
Un message digest (MD) es una función de háshing que toma una longitud arbitraria de texto plano y con el computa una cadena de longitud fija de bits. Esta función debe cumplir con cuatro propiedades:

- Dado un P, es fácil calcular MD(P)
- Dado un MD(P) es efectivamente imposible encontrar P
- Dado un P nadie puede encontrar un P' tal que MD(P') = MD(P)
- El cambio de al menos un bit en el input produce una salida totalmente diferente.

Una vez aplicado la función de messege digest al texto el resultado obtenido es firmado (desencriptándolo con la clave privada propia) y enviado junto con el texto plano.

El receptor para comprobar que la firma sea la adecuada lo que debe realizar es aplicar la misma función de messege digest al mensaje y encriptar con la clave publica del emisor el digest recibido. Luego compara ambos messege digest y si son iguales confirma que quien firmo es quien dice ser y que el mensaje no ha sido modificado.

Los messege digest más utilizados son MD5 propuesto por Rivest en 1992 y SHA-1 propuesto por el NIST en 1993.

## *Protocolos en comunicaciones criptográficas.*

En toda comunicación entre personas se respeta algún tipo de protocolo. Este protocolo incluye el idioma a utilizar, la forma (no es lo misma hablar con el verdulero que con un profesor) y otros aspectos mas. De igual forma la comunicación entre maquinas tiene un protocolo fijo y especificado que le dice como debe interactuar con la otra parte y cuando hablamos de intercambio de mensajes criptográficos tanto entre maquinas como entre personas se debe utilizar un protocolo especifico que establece que se debe hacer, cuando y como.

Tan importante como el método de encriptación en si es el protocolo a utilizar en el intercambio. Se puede observar que ciertos métodos condicionan a los protocolos y viceversa.

Pero antes de continuar es importante definir que entendemos por protocolo:

**Protocolo**
Un protocolo es una serie de pasos, que involucra a dos o más partes, designados para resolver una tarea. Existe una secuencia desde el inicio hasta el final. Cada paso debe ejecutarse en su turno, y no se pueden tomar otros pasos hasta que el anterior se haya finalizado.

Los protocolos tienen otras características como:

- Todos los involucrados en el protocolo deben conocerlo y todos los pasos a seguir a priori.
- Todos los involucrados en el protocolo deben estar de acuerdo en seguirlo
- El protocolo no debe ser ambiguo, cada paso debe estar bien definido y no debe existir chance de desentendimiento.
- El protocolo debe ser completo. Debe existir una acción para cada una de las posibles situaciones.
- No debe ser posible hacer más o conocer más de lo que dice el protocolo.

Los protocolos criptográficos involucran en su funcionamiento el uso de criptografía. Los objetivos de los protocolos criptográficos son variados y van desde el intercambio de información en forma secreta, hasta verificar la identidad de alguien, asegurar la autenticidad y la invariabilidad de un texto, generar un número aleatorio para un juego y otras muchas aplicaciones más.
Se pueden clasificar a los protocolos por la gente que involucra:

- **Protocolos arbitrados**: Además de las 2 partes básicas en el protocolo existe una tercera persona que se encarga de validar y verificar que las partes estén cumpliendo correctamente sus pasos, que no se realicen violaciones al

protocolo y que certifica que todo se realizo en forma normal. Para esto se ocupa de intermediario entre las acciones de una parte y la otra.
- **Protocolos adjudicados**: En este caso las 2 partes actúan una con la otra en forma directa y en caso de disputa o desacuerdo involucran a un tercero que en base a las evidencias presentadas por cada parte dicta una decisión.
- **Protocolos autosuficientes**: En este caso el protocolo de por si garantiza el cumplimiento adecuado de cada etapa por cada una de las partes. No se necesita en ningún momento un tercero y el método de por si es 100% confiable.

En primera instancia se analizaran los protocolos tendientes a realizar un intercambio de mensajes en forma segura entre dos partes, luego se analizaran protocolos destinados.

Para todos los ejemplos que se darán se utilizaran el nombre de personas que ya se han vuelto un clásico en el mundo de la criptografia: Bob y Alice son las personas interesadas en realizar una comunicación. Trent será un árbitro en el caso de ser requerido por el protocolo.

## Comunicaciones con criptosistemas simétricos

Para lograr la comunicación segura entre dos partes utilizando criptografía simétrica se utiliza el siguiente protocolo

1. Alice y Bob se ponen de acuerdo sobre el criptosistema a utilizar
2. Alice y Bob acuerdan una clave
3. Alice encripta el mensaje usando la clave y lo envía a Bob
4. Bob desencripta el mensaje con la clave y lee el mensaje

En este protocolo se debe tener en cuenta:

- El criptosistema a utilizar se puede definir en público. En cambio la clave a utilizar debe definirse en forma privada.
- Una persona malintencionada si conoce la clave o rompe el criptosistema puede interceptar el mensaje, leerlo e incluso cambiarlo sin que las partes se enteren en todo el procedimiento.
- Bob y Alice deben tenerse mutuamente confianza. Alice no va a desconocer que envió el mensaje, y Bob no va a desconocer que lo recibió.
- El protocolo no involucra a terceros

## Comunicaciones con criptosistemas asimétricos

Para lograr la comunicación entre dos partes utilizando criptografía asimetrica se deben realizar los siguientes pasos:

1. Alice y Bob se ponen de acuerdo sobre el criptosistema a utilizar
2. Bob envía a Alice su clave pública
3. Alice encripta el mensaje con la clave pública de Bob y se lo envía
4. Bob desencripta el mensaje con su clave privada y lo lee

Se puede ver que:

- No se requiere un intercambio secreto de claves. El receptor envía su clave publica para que el receptor pueda encriptar el mensaje con esta
- Únicamente quien tenga la clave privada puede desencriptar el mensaje (El receptor).
- Podría ocurrir que un tercero intercepte el mensaje y lo reemplace por otro encriptandolo con la clave pública del receptor sin que Bob o Alice se den cuenta.
- El protocolo no involucra a terceros

## Comunicaciones con criptosistemas híbridos

Cuando se habla de comunicaciones con criptosistemas híbridos se refiere a la utilización de protocolos que utilizan tanto criptosistemas simétricos y asimétricos.

La criptografía simétrica tiene como talón de Aquiles el hecho de tener que distribuir la clave a utilizar con anterioridad y que esta clave permite tanto comprimir como descomprimir. Por otro lado la criptografía asimétrica es en general lenta y tarda en encriptar y desencriptar en general 1000 veces más que la simétrica. Además requiere un ancho de banda mayor a la hora de enviar el criptograma.

En pocos pasos:

1. Bob envía a Alice su clave pública
2. Alice genera una clave temporal, la encripta utilizando la clave pública de Bob y se la envía
3. Bob desencripta la clave temporal
4. Ambos encriptan sus mensajes durante la sesión utilizando la clave temporal

De este protocolo se desprende:

- Se utiliza un criptosistema de clave pública para encriptar la clave a utilizar en un criptosistema de clave privada.
- La clave privada a utilizar es desechable y en otra conversación se debe utilizar otra.
- La clave temporal puede ser encriptada en forma rápida por un criptosistema de clave pública pues es de una longitud corta.
- Se soluciona la problemática del intercambio de la clave del criptosistema simétrico.

## Firma digital con criptosistemas simétricos

A primera vista parece complicado realizar un protocolo que Como se ve en la comunicación no hay forma de comprobar que quien mando el mensaje es realmente quien dice ser. Para eso se puede agregar un árbitro en el medio de la comunicación que certifique que el mensaje enviado realmente fue enviado por quien dice serlo.
El protocolo en este caso sería el siguiente:

1. Alice encripta su mensaje para Bob con una clave Ka y la envía a Trent
2. Trent desencripta el mensaje con Ka
3. Trent une el mensaje a una certificación de que recibió el mensaje de parte de Alicia, encripta todo esto con Kb, y se lo envía a Bob

4. Bob desencripta lo recibido con Kb. Lee el mensaje y la certificación de Trent.

De este protocolo se desprende:

- Se utiliza dos juegos independientes de claves, una para el emisor y el árbitro y otra para el árbitro y el receptor. La determinación de las claves debe mantenerse nuevamente en absoluto secreto y puede ser definida mucho antes de la necesidad de la comunicación y reutilizada una cantidad prudencial de veces.
- El árbitro debe ser alguien de confianza para los dos (como todo árbitro debería ser).
- Si una clave es conocida por un tercero, en cualquiera de las dos comunicaciones puede ocasionar que el mensaje se altere.
- En el caso de que Alice y Bob no se tengan mutua confianza, Trent puede certificar que lo que le llego a Bob es lo que recibió de Alice.
- Se puede decir que el mensaje enviado esta **firmado**. Y esta es una aplicación de firma digital utilizando únicamente criptografía simétrica

## Firma digital con criptosistemas asimétricos

Los sistemas criptograficos asimétricos permitieron el auge de la firma digital. Su utilización en forma simple y elegante permite resolver el problema de la firma digital en pocos pasos:

1. Alice desencripta el mensaje con su clave privada (firma)
2. Alice envía el mensaje a Bob
3. Bob encripta el mensaje con la clave pública de Alice, verificando la firma

Se puede ver que:

- No se requiere la intervención de un tercero
- La firma se realiza con la clave privada propia que puede ser verificada únicamente con la clave pública del firmante. Si Alice al tratar de encriptar para verificar la firma no puede hacerlo significa que el mensaje tiene una firma no valida.
- La firma cumple con ser autentica, inimitable (solo Alice tiene su clave privada), no puede ser reutilizable (el resultado de la firma es en función del documento), el documento es inalterable (si se modifica no podrá ser procesado correctamente y por lo tanto deja de ser valido) y la firma no puede ser repudiada.

## Firma digital con message digest

La utilización de message digest junto con la criptografía asimétrica permite la firma de mensajes donde la firma se encuentra físicamente separada al documento y aun así el mensaje permanece probando la validez únicamente del mismo.
Se deben realizar los siguientes pasos:

1. Alice y Bob se ponen de acuerdo sobre el criptosistema y función de Hash a utilizar.
2. Alice procesa el mensaje con la función de Hash (MD)
3. Alice firma (desencripta con su clave privada) el resultado del MD

4. Alice envia a BoB el mensaje y la firma
5. Bob procesa el mensaje recibido con la función de Hash (MD)
6. Bob encripta la firma recibida con la clave publica de Alice
7. Bob verifica que sean iguales el MD del mensaje recibido y el obtenido luego de procesar la firma.

Se puede ver que:

- La firma autentica al emisor
- El mensaje en toda la comunicación nunca es encriptado.
- El mecanismo es autosuficiente y no se requieren terceros.
- Al firmarse el message digest y no el mensaje entero el proceso es más rápido (El message digest es mas pequeño que el mensaje).

## Comunicaciones con criptosistemas asimétricos y firma.

Para lograr enviar un mensaje de encriptado y firmado simplemente se puede utilizar el protocolo de firma y de encriptación simultáneamente, teniendo en cuenta el orden de las cosas:

1. Alice firma el mensaje con su clave privada
2. Alice encripta el mensaje con la clave publica de Bob
3. Alice envia el mensaje a Bob.
4. Bob recibe el mensaje y lo desencripta con su clave privada.
5. Bob verifica la firma con la clave pública de Alice.

## Bit-Commitment

En este caso lo que se quiere es comprometerse a un resultado pero no revelar la predicción hasta tiempo después. Ejemplo de esto puede ser pronosticar un resultado en una competencia deportiva pero no haciéndolo publico para no influir en las apuestas. Una vez pasado el evento deseo que se pueda comprobar fehacientemente mi acierto (o error) sin que queden dudas sobre la verosimilitud de mi predicción (que nadie pueda decir que se cambio la predicción).

Para esto se puede utilizar un simple protocolo utilizando un criptosistema simétrico:

1. Bob Genera un una cadena aleatoria de bit “R” y se la envía a Alice
2. Alice crea un mensaje un la cadena “R” y la cadena de bits “B” con su pronostico (o compromiso). Luego encripta el resultado con una clave privada “K” y envía el resultado a Bob

Luego de un tiempo cuando se quiera observar la predicción:

3. Alice envía la clave privada “K”
4. Bob desencripta el mensaje para ver la predicción. Y chequea la cadena “R” para ver si es la misma que el mando en primer lugar.

Si el mensaje no contiene la cadena random “R” Alice podría buscar claves diferentes que al desencriptar el mensaje produzcan el resultado deseado. Al agregar la cadena

random este proceso tendrá que lograr esto sin que cambie esta cadena. Si el método de encriptación es bueno la posibilidad de que esto ocurra es mínima.

## Fair Coin Flips

La acción de tirar una moneda tratar de adivinar que saldrá (cara o seca) es utilizado ampliamente: se lo puede ver al inicio de los partidos de fútbol donde quien gana puede elegir el arco o si comienza el partido sacando o en la toma de decisiones trascendentales tales como “si sale cara encaro a esa chica (o ese chico)”. Teniendo una moneda eso es sencillo pero tener que realizar esa acción mediante una computadora y a través de una red eso no es tan sencillo. Para lograr esto se debe contar con un protocolo que:

- Se tire la moneda antes que el otro se comprometa a un resultado (adivine).
- No se pueda tirar la moneda nuevamente después de comprometerse
- Quien adivine no debe ser capaz de ver el resultado de la moneda lanzada.

Una solución sencilla involucra el protocolo de bit-commitment y es la siguiente:

1. Alice selecciona un bit aleatorio (cara o seca) y utiliza el protocolo de bit-commitment para protegerlo
2. Bob trata de adivinar el resultado
3. Alice revela el resultado a Bob (resolviendo el protoco bit-commitment). Bob gana si acertó.

## Zero-Knowledge Proofs

Puede existir cierta ocasión donde debemos demostrar cierto conocimiento nuestro a sobre un tema en especial sin darle el conocimiento en si. Por ejemplo:

A: “conozco la combinación de la caja fuerte donde se guardan los libros de actas de la facultad”

B: “No te creo”

A: “Si, en serio”

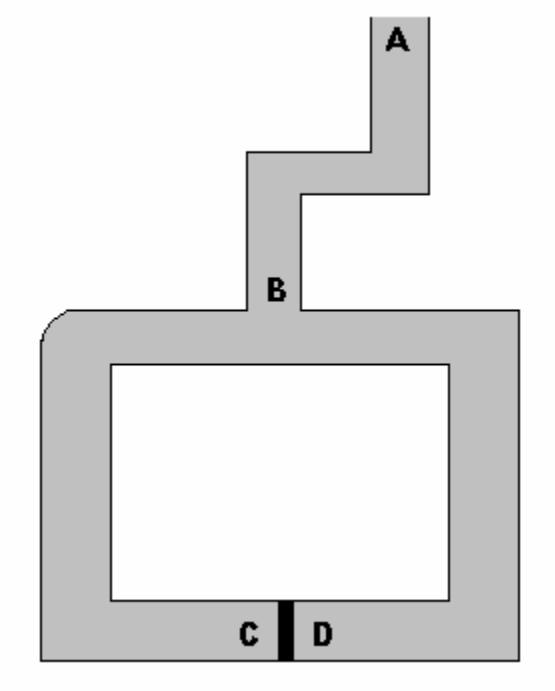
B: “¡Probalo!”

A: “El número es ###...#” (censurado)

B: “Que bueno, ahora que lo se voy a cambiar todas mis notas…”

Para evitar este tipo de situaciones se debe lograr mediante algún protocolo demostrar mi conocimiento, convencer al otro que lo tengo, pero no revelarle cual es el mismo.
Este protocolo es conocido como el Zero-knowledge proofs y es un método iterativo.
La explicación del mismo se da en base a un ejemplo típico utilizado ampliamente en la bibliografía: “El problema de la cueva” y es el siguiente.

Imaginemos la existencia de una cueva como la mostrada en la figura, donde existe una puerta con una combinación secreta que permite pasar de un lado al otro de la misma.




Además una persona le quiere demostrar a otro que conoce la combinación sin decirselo. Para esto se ponen de acuerdo en lo siguiente:

1. Bob que no conoce la clave se va a poner en el punto “A”
2. Alice (que conoce la clave) entra en la cueva y va hasta el punto “C” o el “D”
3. Después que Alice entro, Bob se para en el punto “B”
4. Bob le grita a Alice que salga:
    a. Por el lado derecho o
    b. Por el lado izquierdo
5. Alice cumple lo pedido utilizando la combinación para pasar por la puerta (si es necesario)
6. Los pasos 1 a 5 se repiten “n” veces.

Si en realidad Alice no conoce la clave puede ocurrir que entre al punto “C” o “D” y no necesite abrir la puerta para salir por el lado pedido, podría tener suerte. (50% de posibilidad). Por esto es que el método se repite tantas veces como sea necesario para que Bob este seguro que Alice realmente tenga el conocimiento y no mucha suerte. Basta con que en una iteración Alice no salga por el lado pedido para determinar que no conoce la combinación. Por otro lado nunca estaremos 100% seguro que conoce la combinación pero podemos llegar hasta el grado de certeza que deseemos (100, 1000, etc. iteraciones).

## Digital Cash

Cada día se hace más común comprar productos por internet. Encontrar una forma de realizar transacciones electrónicas seguras es un reto en si. Hasta el momento la forma mas difundida de hacerlo es mediante la compra con tarjeta de crédito, pero la tarjeta de crédito a diferencia del dinero identifica a la persona que compro el producto. Existe gente que valora su privacidad y este problema les parece un escollo insuperable. Es por eso que solicitan la existencia de un método alternativo de pago electrónico y que funcione como el dinero en papel.

Para lograr esto encuentran deseable las siguientes características:

- Independencia: La seguridad del efectivo digital no debe ser dependiente de ninguna locación física. Este deberá poder ser transferido por redes de computadoras.
- Seguridad: No debe poder ser copiado y reusado.
- Privacidad: No debe poder ser reastreado las relaciones entre un usuario y sus compras.
- Pago Off-line: Pagando con Digital Cash no es necesario estar enlazado a un host para procesar el pago
- Transferencia: El Digital Cash debe poder ser transferido a otro usuario.
- Divisibilidad: Una pieza de Digital Cash de un monto dado debe poder ser subdividido en piezas de montos menores.

Se detalla a continuación un método que cumple algunos de las características anteriores que por su simplicidad permite ser entendido sin mayores complicaciones.

1. Alice prepara 100 órdenes anónimas de dinero por $1000 cada una
2. Alice pone en cada una de ellas junto con un pedazo de papel carbónico en 100 sobres diferentes. Entrega los mismos al banco
3. El banco abre 99 sobres y confirma que cada orden es por $1000
4. El banco firma el sobre que quedó cerrado. La firma pasa del sobre mediante el papel carbónico a la orden de compra. El banco le retorna el sobre a Alice y descuenta $1000 de su cuenta.
5. Alice abre el sobre y gasta la orden de compra en un comercio
6. El comerciante verfica la firma del banco para verificar que la orden sea legitima.
7. El comerciante lleva la orden al banco
8. El banco verifica su firma y acredita $1000 en la cuenta del comerciante.

Este protocolo que se realiza persona a persona puede ser reemplazado por uno de procesamiento digital. La firma del banco pasará a ser una firma digital. Para la utilización de las 100 ordenes anónimas y la apertura de 99 se puede utilizar el algoritmo de Zero-Knowledge Proofs. El sobre puede ser reemplazado por la encriptación de las órdenes y de esa forma lograr que este sea un método aplicable utilizando medios electrónicos.

Este método logra independencia, privacidad y pago off-line. Para agregar seguridad simplemente se debe agregar en cada orden de compra una cadena aleatoria de una longitud larga (cada orden debe tener otra cadena). Cuando el banco recibe la orden anónima por primera vez anota la cadena en una lista. De esta forma si la orden trata de depositarse una vez más será rechazada.

El mecanismo presentado tiene varios puntos discutibles y por mejorar. Sin embargo sirve para entender el funcionamiento basico del Digital Cash.

T. Okamoto y K. Ohta en su publicación “Universal Electronic Cash” en el año 1992 proponen un mecanismo que cumple con todos los características deseables del Digital Cash.

## *Algunas anécdotas Criptográficas*

### La X marca el lugar…

La historia de la criptografía está llena de anécdotas y personajes pintorescos. David Kahn narra en "The Codebreakers" (MacMillan 1967), entre otras, la historia de Thomas Jefferson Beale, un aventurero que en 1821, emprendió un arriesgado viaje. Antes de partir, dejó a Robert Morris su huésped, dos documentos cifrados para que los abriese si él desaparecía. Morris se decidió a hacerlo al cabo de 20 años. Con la ayuda de unos amigos, descifró el segundo texto: era una sucesión de números comprendidos entre el 1 y el 1322, que correspondía a la Declaración de Independencia. Sustituyendo cada número por las iniciales de las palabras de la Declaración, se obtenía la descripción de un tesoro de una tonelada y media de oro, dos y media de plata y gran cantidad de joyas. El lugar en que se encontraba enterrado estaba indicado con precisión en el primer texto. Es una secuencia de 520 números comprendidos entre el 1 y el 2906, pero nadie ha descubierto en base a qué documento están numeradas las palabras, lo que permitiría descifrar el mensaje. Durante un siglo y medio se ha intentado relacionarlo con la Constitución estadounidense y con la Biblia, entre otros textos, y se han realizado costosos análisis por ordenador, sin resultados.

Quizá sólo sea una broma pesada, pero por si acaso, aquí están los 520 números:

71, 194, 38, 1701, 89, 76, 11, 83, 1629, 48, 94, 63, 132, 16, 111, 95, 84, 341, 975, 14, 40, 64,
27, 81, 139, 213, 63, 90, 1120, 8, 15, 3, 126, 2018, 40, 74, 758, 458, 604, 230, 436, 664, 582,
150, 251, 284, 308, 231, 124, 211, 4866, 225, 401, 370, 11, 101, 305, 139, 189, 17, 33, 88,
208, 193, 145, 1, 94, 73, 416, 918, 263, 28, 500, 538, 356, 117, 136, 219, 27, 176, 130, 10,
460, 25, 485, 18, 436, 65, 84, 200, 283, 118, 320, 138, 36, 416, 280, 15, 71, 224, 961, 44, 16,
401, 39, 88, 61, 304, 12, 21, 24, 283, 134, 92, 63, 246, 486, 682, 7, 219, 184, 360, 780, 18,
64, 463, 474, 131, 160, 79, 73, 440, 95, 18, 64, 581, 34, 69, 128, 367, 460, 17, 81, 12, 103,
820, 62, 116, 97, 10, 862, 70, 60, 1317, 471, 540, 208, 121, 890, 346, 36, 150, 59, 568, 614,
13, 120, 63, 219, 812, 2160, 1780, 99, 35, 18, 21, 136, 872, 15, 28, 170, 88, 4, 30, 44, 112,
18, 147, 436, 195, 320, 37, 122, 113, 6, 140, 8, 120, 305, 42, 58, 461, 44, 106, 301, 13, 408,
680, 93, 86, 116, 530, 82, 568, 9, 102, 38, 416, 89, 71, 216, 728, 965, 818, 2, 38, 121, 195,
14, 326, 148, 234, 18, 55, 131, 234, 361, 824, 5, 81, 623, 48, 961, 19, 26, 33, 10, 1101, 365,
92, 88, 181, 275, 346, 201, 206, 86, 36, 219, 320, 829, 840, 68, 326, 19, 48, 122, 85, 216,
284, 919, 861, 326, 985, 233, 64, 68, 232, 431, 960, 50, 29, 81, 216, 321, 603, 14, 612, 81,
360, 36, 51, 62, 194, 78, 60, 200, 314, 676, 112, 4, 28, 18, 61, 136, 247, 819, 921, 1060, 464,
895, 10, 6, 66, 119, 38, 41, 49, 602, 423, 962, 302, 294, 875, 78, 14, 23, 111, 109, 62, 31,
501, 823, 216, 280, 34, 24, 250, 1000, 162, 286, 19, 21, 17, 340, 19, 242, 31, 86, 234, 140,
607, 115, 33, 191, 67, 104, 86, 52, 88, 16, 80, 121, 67, 95, 122, 216, 548, 96, 11, 201, 77,
364, 218, 65, 667, 890, 236, 154, 211, 10, 98, 34, 119, 56, 216, 119, 71, 218, 1164, 1496,
1817, 51, 39, 210, 36, 3, 19, 540, 232, 22, 141, 617, 84, 290, 80, 46, 207, 411, 150, 29, 38,
46, 172, 85, 194, 36, 261, 543, 897, 624, 18, 212, 416, 127, 931, 19, 4, 63, 96, 12, 101, 418,
16, 140, 230, 460, 538, 19, 27, 88, 612, 1431, 90, 716, 275, 74, 83, 11, 426, 89, 72, 84, 1300,
1706, 814, 221, 132, 40, 102, 34, 858, 975, 1101, 84, 16, 79, 23, 16, 81, 122, 324, 403, 912,
227, 936, 447, 55, 86, 34, 43, 212, 107, 96, 314, 264, 1065, 323, 328, 601, 203, 124, 95, 216,
814, 2906, 654, 820, 2, 301, 112, 176, 213, 71, 87, 96, 202, 35, 10, 2, 41, 17, 84, 221, 736,
820, 214, 11, 60, 760.

**El concurso de KNAPSACKS.**

El inventor del algoritmo de Knapsacks, Ralph Merkle, estaba tan seguro de que el mismo no podría quebrarse que ofreció una recompensa de $100 dólares a quien pudiese hacerlo. Inmediatamente Adi Shamir quebró el código y cobró la recompensa. No dejándose intimidar por esto Merkle fortificó el método y ofreció $1000 dólares de recompensa a quien pudiese quebrar al nuevo algoritmo. Ronald Rivest rápidamente cumplió con este nuevo desafío y obtuvo el pago. Merkle no se animó a ofrecer una recompensa de $10.000 para su nueva versión. De todas formas el algoritmo Knapsacks no es considerado seguro y no es usado en la práctica.

Rivest, Shamir y Leonard Adleman en 1978 propusieron un método hoy ampliamente utilizado conocido como RSA (Conocido así por las tres iniciales de sus apellidos). Al parecer Adleman no tuvo la suerte de sus compañeros y se quedo esperando por la recompensa de $10.000 que Merkle nunca ofreció.

**DES(cifrado)**

El Estándar Federal para encriptación de datos. (DES) fue durante mucho tiempo un buen algoritmo de encriptación para la mayoría de las aplicaciones comerciales. El gobierno de USA, sin embargo nunca confío en el DES para proteger sus datos clasificados debido a que la longitud de la clave del DES era de solamente 56 bits, lo suficientemente corta como para ser vulnerable a un ataque por fuerza bruta.

El ataque mas devastador contra el DES fue descripto en la conferencia Crypto'93 donde Michael Wiener de Bell presento un trabajo sobre como crackear el DES con una maquina especial. El diseño consistía en un Chip especial que probaba 50 millones de claves DES por segundo hasta que encontraba la correcta, estos chips podían producirse por $10.50 cada uno, y Wiener había desarrollado una maquina especial que reunía 57000 de estos chips a un costo de un millón de dólares. La maquina era capaz de crackear cualquier clave DES en menos de siete horas promediando 3.5 horas por clave. Por 10 millones Wiener construía una maquina que tardaba 21 minutos por clave. Y por 100 millones el tiempo se reducía a dos minutos por clave. Desde ese momento el DES de 56 bits no volvió a ser utilizado con propósitos serios de encriptación de datos.

Curiosamente, pocas semanas antes, un alto cargo de la NSA había declarado que dicho algoritmo seguía siendo seguro, y que descifrar un mensaje resultaba aún excesivamente costoso, incluso para organizaciones gubernamentales.

A pesar de su caída, DES siguió siendo ampliamente utilizado en multitud de aplicaciones, como por ejemplo las transacciones de los cajeros automáticos. De todas formas, el problema real de DES no radica en su diseño, sino en que emplea una clave demasiado corta (56 bits), lo cual hace que con el avance actual de las computadoras los ataques por la fuerza bruta comiencen a ser opciones realistas.

**Factorizando primos...**

La dificultad de la factorización de grandes números múltiplos de números primos es aprovechada en la encriptación en métodos de clave pública. No hay un algoritmo que resuelva este problema en forma rápido y requiere gran esfuerzo computacional.

Allá por el siglo 17, Cataldi, un matemático de renombre de la época, trabajaba intentando descubrir nuevos números primos. Cataldi había encontrado tres números de unos 200 dígitos de los cuales sospechaba que eran primos, como no podía probarlo, le envío los tres números al ilustre Fermat.

Fermat: Fermat fue un matemático sumamente misterioso, su teorema más famoso denominado “el último teorema de Fermat” era aquel que decía que para n>2 no existían enteros a,b,c tales que a^n = b^n + c^n. Fermat anotó en su cuaderno que había encontrado una demostración muy elegante para este teorema pero que lamentablemente no entraba en el margen del cuaderno. Nunca más se supo de otro manuscrito de Fermat sobre el tema y el teorema permaneció inconcluso hasta 1995, año en el cual dos científicos ingleses presentaron un mamotreto de 200 paginas con una supuesta demostración, la cual está aún en estudio!!.

La cuestión es que el misterioso Fermat le envía a Cataldi dos meses más tarde una carta en la cual le decía que los dos primeros números si eran primos (felicitaciones Cataldi !) pero el tercero no lo era pues era divisible por dos números, los cuales sí eran primos (Fermat le envió los números a Cataldi). Hasta el día de hoy no hay algoritmo ni computadora que sea capaz de factorizar un numero de 200 dígitos en dos meses. Y Fermat no tenía computadoras!!!!!

Como puede observarse el ataque mas devastador contra el RSA ocurrió en el siglo XVII y hasta el día de la fecha no se ha vuelto a repetir. Se supone que el misterioso Fermat se ha llevado a la tumba un método para factorizar números que hasta el día de hoy nadie ha descubierto. Mientras no aparezca otro Fermat el RSA permanece tranquilo.

**Sin agua potable**

Los códigos de la versión japonesa de Enigma (llamados Purple, violeta) se descifraron en el atolón de Midway. Un grupo de analistas, dirigidos por el comandante Joseph J. Rochefort, descubrió que los nipones señalaban con las siglas AF su objetivo. Para comprobarlo, Rochefort les hizo llegar este mensaje: "En Midway se han quedado sin instalaciones de desalinización". Inmediatamente, los japoneses la retransmitieron en código: "No hay agua potable en AF". De esta forma, el almirante Nimitz consiguió una clamorosa victoria, hundiendo en Midway cuatro portaviones japoneses.

**Idiomas secretos**

Los sacerdotes egipcios tenían un alfabeto secreto, el demótico que utilizaban en sus escrituras. El pueblo, en cambio, utilizaba el alfabeto Hierático. El secreto del mismo pasaba de generación en generación.

En la segunda guerra mundial los Estados Unidos utilizaron para transmitir sus mensajes secretos a indígenas de la tribu navajo. Estos individuos transmitían utilizando su idioma nativo. El método fue utilizado con éxito y nunca fueron violados los secretos transmitidos.

Hace años en china estaba prohibido para cualquier mujer aprender a leer y escribir. No obstante un grupo de mujeres invento su propio método de escritura que guardaron celosamente. Cancelada la prohibición el alfabeto cayó paulatinamente en el olvido

**Enigma**

En el año 1923, un ingeniero alemán patentó una máquina específicamente diseñada para facilitar las comunicaciones seguras. Se trataba de un instrumento parecido a una máquina de escribir. La encriptación se obtenía gracias a las posiciones iniciales de tres tambores o rotores que el ingenio poseía en su parte frontal. Los rotores no son más que tambores con contactos en su superficie y cableados en su interior,
de forma que con cada pulsación del teclado, la posición de estos determina cuál es la letra
que se ha de iluminar. Cada vez que se pulsa una tecla el primer rotor avanza una posición;
el segundo avanza cuando el anterior ha dado una vuelta completa y así sucesivamente.

Aunque ENIGMA parecía virtualmente imposible de romper, presentaba una serie de debilidades, tanto en su diseño como en los mecanismos empleados para utilizarla,

El primero en conseguir avances significativos fue el servicio de inteligencia polaco, ya que en 1931 los franceses, en virtud de un acuerdo de cooperación firmado diez años antes, les facilitaron información detallada sobre la máquina, que ellos a su vez habían obtenido sobornando a un miembro de la oficina de cifras alemana. De hecho, los espías franceses consideraban esta información totalmente inútil, ya que pensaban que ENIGMA era, sencillamente, indescifrable.

El conocimiento preciso de la máquina permitió a un equipo de tres matemáticos elaborar un aparato “bomba” que permitía descifrar los mensajes aprovechando una debilidad, no en la máquina en sí, sino en el protocolo empleado por el ejercito alemán para colocar los rotores al principio de cada mensaje.

En 1938 Alemania cambió el protocolo e introdujeron dos rotores adicionales, quedando inservible el decodificador creado.

Con la caída de Polonia, la posta de investigación paso a Inglaterra. Un grupo de matemáticas ingleses, liderados por Alan Turing desarrollaron una segunda “Bomba” basándose en los estudios del polaco, más evolucionada y rápida que su antecesora Este nuevo dispositivo aprovechaba una debilidad esencial en ENIGMA: un mensaje no puede codificarse en sí mismo, lo cual implica que ninguna de las letras del texto claro puede coincidir con ninguna del texto cifrado. La Bomba de Turing partía de una palabra adivinada —en contra de las normas de uso de ENIGMA, la mayoría de los mensajes que enviaba el ejército alemán comenzaban de igual forma, lo cual facilitó la tarea del equipo aliado enormemente—, y buscaba un emparejamiento con el mensaje cifrado tal que el supuesto texto claro y el fragmento de criptograma asociado no coincidieran en ninguna letra. A partir de ahí la Bomba realizaba una búsqueda exhaustiva de la configuración inicial de la máquina para decodificar el mensaje.

## *Bibliografía.*

- Bruce Schneier, “Applied Cryptography 2d”, 1996.
- Manuel José Lucena Lopez, “Criptografía y Seguridad en Computadores, Tercera Edición”
- Menezes - van Oorschot – Vanstone, “Handbook of applied Cryptography”, 1997
- David Kahn, “The Codebreakers”, MacMillan, 1967
- PGP User's Manual.
- PGP advanced user's information.
- Arto Salomaa, “Public Key Criptography”.
- R.L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, 1977
- Tanenbaum, “Computer Networks, Fourth Edition”, 2003
- Bruce Schneier, “A Self-Study Course in Block-Cipher Cryptanalysis”
- T. Okamoto y K. Ohta, “Universal Electronic Cash,” Advances in Cryptology—CRYPTO ’91 Proceedings, Springer–Verlag, 1992, pp. 324–337.

# *Índice Alfabético*


## A

Ataque por diccionario....21

## B

Bit-Commitment....36
Bloques, encriptación por....16

## C

Caballos de Troya....21
Caesar, Criptosistema....9
Cifrado monoalfabético general....10
Cipher Block Chainning....16
Cipher Feedback....17
Computacionalmente seguro....6
Criptoanálisis....6
Criptografía....4
Criptografía asimétrica....22
Criptografía de clave privada,....*Véase* Criptografía simétrica
Criptografia de clave pública....*Véase* Criptografia asimétrica
Criptografía simétrica....7
Criptograma....5
Criptosistema....5

## D

DES....18
Digital Cash....38

## E

Electronic Code Block....16
Escitalo....8

## F

Fair Coin Flips....37
Firma digital....30
Fuerza Bruta....21

## H

Hill, Criptosistema....13
Homofónicos....8

## I

incondicionalmente seguro....6
Ingeniería Social....21

## K

KNAPSACKS....23

## M

Message digest....31
Monoalfabeticos....8

## O

one-time pad....7
Output Feedback....17

## P

PGP....28
Plaintext....*Véase* Texto plano
PlayFair, Método....11
Polialfabéticos....8
Poligráficos....8
Protocolo....32
Protocolos adjudicados....33
Protocolos arbitrados....32
Protocolos autosuficientes....33

## R

Relativamente primos....10
RSA....26

## S

Shoulder Surfing....21
Sustitución....8
Sustitución afín, Criptosistema....9

## T

Texto plano....5
Transposición....8
Triple DES....18
Troya, Caballos de....21

## V

Vector super incrementante....24
Vigènere, cifrado....15

## Z

Zero-Knowledge Proofs....37