FACULTAD DE INGENIERIA

Introducción a la criptografía

75.06 - Organización de Datos

FACULTAD DE INGENIERIA UNIVERSIDAD DE BUENOS AIRES

Departamento de Computación

NOVIEMBRE 2004

Índice General

1 Conceptos básicos sobre Criptografía

1.1 Criptografía

La palabra criptografía proviene del griego y significa “escritura secreta”. Ésta tiene una larga y pintoresca historia que proviene de hace miles de años. Su definición es: “Arte de escribir con clave secreta o de un modo enigmático”. Obviamente la criptografía hace años que dejó de ser un arte para convertirse en una técnica, o más bien un conglomerado de técnicas, que tratan sobre la protección (ocultamiento frente a observadores no autorizados) de la información. Entre las disciplinas que engloba cabe destacar la Teoría de la Información, la Teoría de Números (estudio de las propiedades de los números enteros) y la Complejidad Algorítmica.

Formalmente definiremos a la criptografía como el estudio de técnicas matemáticas relacionadas con aspectos de seguridad de la información como confidencialidad, integridad de los datos, autentificación de entidad y autentificación del origen de los datos. A continuación se detallan cada uno de los objetivos de la criptografía:

- **Confidencialidad**: es un servicio utilizado para mantener el contenido de la información para todos aquellos que estén autorizados a tenerla. *Secreto* es un sinónimo de confidencialidad y privacidad.
- **Integridad de los datos**: Se dice que la integridad de estos datos ha sido preservada cuando los datos no han sido alterados (modificados, borrados) de una manera no autorizada desde el momento en que fueron creados, transmitidos o guardados por una fuente autorizada. Para poder asegurar la integridad de los datos, se requiere la habilidad de detectar su manipulación por quien no posee la autoridad para hacerlo. La manipulación o alteración de los datos incluye inserción, borrado o sustitución de partes o del todo.
- **Autentificación**: es un servicio asociado a la identificación. Por autentificación entenderemos cualquier método que nos permita comprobar de manera segura alguna característica sobre un objeto. Dicha característica puede ser su origen, su integridad, su identidad, etc.
- **No repudio:** se trata de que una vez enviado un mensaje, el emisor no pueda negar haber sido el autor de dicho envío. El no repudio es condición suficiente para la autenticidad, por lo que si un mensaje es no repudiable es auténtico, pero no al revés.

Existen dos documentos fundamentales, uno escrito por Clause Shannon en 1948 ("A Mathematical Theory of Communication"), en el que se sentaban las bases de la Teoría de la Información, y que junto con otro artículo posterior del mismo autor sirvió de base para la Criptografía moderna. El otro trabajo, publicado por Whitfield Diffie en 1975, introducía el concepto de Criptografía de Clave Pública, abriendo enormemente el abanico de aplicación de esta disciplina.

Históricamente, cuatro grupos de personas han usado y contribuido al arte de la criptografía: los militares, los cuerpos diplomáticos, los escritores y los amantes. De éstos los militares tienen el rol más importante a lo largo de los siglos.

Los profesionales hacen una distinción entre *cifrados* y *códigos*. Un *cifrado* es una transformación carácter-por-carácter o bit-por-bit, sin tener interés en la estructura lingüística del mensaje. En cambio, un *código* reemplaza una palabra con otra palabra o símbolo. Los códigos no se utilizan más en la actualidad, pero tuvieron su momento de gloria en la historia.

1.2 Criptosistema

Definiremos un criptosistema como una quíntupla (*M,C,K,E,D*), donde:

- *M* representa el conjunto de todos los mensajes sin cifrar (lo que se denomina texto plano, o plaintext) que pueden ser enviados.
- *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.
- *E* es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento de C. Existe una transformación diferente E_K para cada valor posible de la clave k.
- *D* es el conjunto de transformaciones de descifrado, análogo a E.

Todo criptosistema ha de cumplir la siguiente condición:

$$D_k\,(E_k(m)) = m$$

es decir, que si tenemos un mensaje *m*, lo ciframos empleando la clase *k* y luego lo desciframos empleando la misma clave, obtenemos de nuevo el mensaje original *m*.

Existen dos tipos fundamentales de criptosistemas:

- *Criptosistemas simétricos o de clave privada*. Son aquellos que emplean la misma clave *k* tanto para cifrar como para descifrar. Presentan el inconveniente de que para ser empleados en canales de comunicación la clave *k* debe estar tanto en el emisor como en el receptor, lo cual nos lleva a preguntarnos cómo transmitir la clave de forma segura.

- *Criptosistemas asimétricos o de clave pública.* Emplean una doble clave (k_{pr},k_{Pu}). k_{pr} se conoce como clave privada y k_{Pu} se conoce como clave pública. Una de ellas sirve para la transformación *E* de cifrado y la otra para la transformación *D* de descifrado. En muchos casos son intercambiables, esto es, si empleamos un para cifrar la otra sirve para descifrar y viceversa. Estos criptosistemas deben cumplir además que el conocimiento de la clave pública k_{Pu} no permita calcular la clave privada k_{pr}. Ofrecen un abanico superior de posibilidades pudiendo emplearse para establecer comunicaciones seguras por canales inseguros (puesto que únicamente viaja por el canal la clave pública que sólo sirve para cifrar), o para llevar a cabo autentificaciones.

En la práctica lo que se emplea es una combinación de estos dos tipos de criptosistemas, puesto que los segundos presentan el inconveniente de ser computacionalmente mucho más costosos que los primeros. Lo que se hace en el mundo real es encriptar los mensajes (largos) mediante algoritmos simétricos, que suelen ser muy eficientes, y luego emplear la criptografía asimétrica para encriptar la clave (que suele ser corta en comparación con los mensajes).

A lo dicho anteriormente se puede agregar que para que un criptosistema sea práctico de utilizar, deberá satisfacer las siguientes propiedades:

1. Para cada función de encriptación E_K y cada función de desencriptación D_K debe ser computacionalmente eficiente.
2. Dado un criptograma no se debe poder ser capaz de determinar la clave K de encriptación.

1.3 Criptoanálisis

El criptoanálisis consiste en comprometer la seguridad de un criptosistema. Esto se puede hacer descifrando un mensaje sin conocer la clave, o bien obteniendo la clave empleada para cifrar algún mensaje. No se considera criptoanálisis el descubrimiento de un algoritmo secreto de cifrado; hemos de considerar por el contrario que el algoritmo de cifrado siempre es conocido.

Existen diferentes formas de atacar un criptosistema:

- **Ataque por fuerza bruta**: si se tiene un mensaje cifrado (criptograma), mediante éste método se probarán todas las claves posibles para obtener el texto plano. Si el conjunto de posibles claves es alto este sistema es inviable. Es decir, se inspecciona la componente *K* del criptosistema. Normalmente a este tipo de ataques no se les suele considerar como una forma de criptoanálisis ya que no busca puntos débiles, sino que únicamente utiliza todas las claves posibles.
- **Ataque por texto plano escogido**: consiste en elegir varios textos planos y obtener sus critogramas asociados. Esto implica tener acceso al dispositivo de encriptación, pero no a la clave de encriptación. O sea, se hace una inspección selectiva de la componente *M* del criptosistema.
- **Ataque a partir de texto plano**: el atacante tiene acceso a textos planos y a sus correspondientes criptogramas.
- **Análisis por frecuencias**: éste tipo de ataque es utilizado para romper sistemas criptográficos simétricos y se base en estudiar la frecuencia con la que aparecen los distintos símbolos en un lenguaje determinado y luego estudiar la frecuencia con la que aparecen en los criptogramas. De esta manera se establece una relación y se puede obtener el texto plano.

La Criptografía no sólo se emplea para proteger información, también se utiliza para permitir su autenticación, es decir, para identificar al autor de un mensaje e impedir que nadie suplante su personalidad. En estos casos surge un nuevo tipo de criptoanálisis que está encaminado únicamente a permitir que elementos falsos pasen por buenos. Puede que ni siquiera nos interese descifrar el mensaje original, sino simplemente poder sustituirlo por otro falso y que supere las pruebas de autentificación.

1.4 Compromiso entre Sistema y Criptoanálisis

La información posee un tiempo de vida, a partir del cual pierde su valor. Los datos sobre la estrategia de inversiones a largo plazo de una gran empresa, por

ejemplo, tienen un mayor período de validez que la exclusiva periodística de una sentencia judicial que se va a hacer pública al día siguiente. Será suficiente, pues, tener un sistema que garantice que el tiempo de vida de la propia información que éste alberga. Esto no suele ser fácil, sobre todo porque no tardará lo mismo un oponente que disponga de una única computadora de capacidad modesta, que otro que emplee una red de supercomputadoras. Por eso también ha de tenerse en cuenta si la información que queremos proteger vale más que el esfuerzo del criptoanálisis que se va a necesitar, porque entonces puede que no esté segura. La seguridad de los criptosistemas se suele medir en términos del número de computadoras y del tiempo necesarios para romperlos, y a veces simplemente en función del dinero necesario para llevar a cabo esta tarea con garantías de éxito.

En cualquier caso hoy por hoy existen sistemas que son muy poco costosos (o incluso gratuitos, como el PGP), y que nos garantizan un nivel de protección tal que toda la potencia de cálculo que actualmente hay en el planeta sería insuficiente para romperlos.

Tampoco conviene depositar excesiva confianza en el algoritmo de cifrado, puesto que en el proceso de protección de la información existen otros puntos débiles que deben ser tratados con un cuidado exquisito. Por ejemplo, no tiene sentido emplear algoritmos con niveles de seguridad extremadamente elevados si luego escogemos *claves o passwords* ridículamente fáciles de adivinar. Una práctica muy extendida por desgracia es la de escoger palabras clave que contengan fechas, nombre de familiares, nombres de personajes o lugares de ficción, etc. Son las primeras que un atacante avispado probaría. Tampoco es una práctica recomendable apuntarlas o decírselas a nadie, puesto que si la clave cae en malas manos, todo nuestro sistema queda comprometido, por buenos que sean los algoritmos empleados.

2 Criptografía clásica

En este capítulo haremos un breve repaso de los mecanismos criptográficos considerados clásicos. Podemos llamar así a todos los sistemas de cifrado anteriores a la II Guerra Mundial, o lo que es lo mismo, al nacimiento de las computadoras. Estas técnicas tienen en común que pueden ser empleadas usando simplemente lápiz y papel, y que pueden ser criptoanalizadas casi de la misma forma. De hecho, con la ayuda de las computadoras, los mensajes cifrados empleando estos códigos son fácilmente descifrables, por lo que cayeron rápidamente en desuso. La transición desde la Criptografía clásica a la moderna se da precisamente durante la II Guerra Mundial, cuando el Servicio de Inteligencia aliado rompe la máquina de cifrado del ejército alemán, llamada ENIGMA.

Todos los algoritmos criptográficos clásicos son simétricos, ya que hasta mediados de los años setenta no nació la Criptografía asimétrica.

2.1 Cifrado por bloque

Los algoritmos de cifrado por bloques toman bloques de tamaño fijo del texto en claro y producen un bloque de tamaño fijo de texto cifrado, generalmente del mismo tamaño que la entrada. La asignación de bloques de entrada a bloques de salida debe ser uno a uno para hacer el proceso reversible y parecer aleatoria.

Para aplicar un algoritmo por bloques es necesario descomponer el texto de entrada en bloques de tamaño fijo. Esto se puede hacer de varias maneras:

- **ECB (Electronic Code Book):** Se parte el mensaje en bloques de *k* bits, rellenando el ultimo si es necesario y se encripta cada bloque. Para descifrar se fragmenta el texto cifrado en bloques de *k* bits y se desencripta cada bloque. Este sistema es vulnerable a ataques ya que dos bloques idénticos de la entrada generan el mismo bloque de salida. En la práctica no se utiliza.
- **CBC (Cipher Block Chaining):** Este método soluciona el problema del ECB haciendo una or-exclusiva de cada bloque de texto en claro con el bloque anterior cifrado antes de cifrar. Para el primer bloque se usa un *vector de inicialización*. Este es uno de los esquemas más empleados en la práctica.
- **OFB (Output Feedback Mode):** Este sistema emplea la *clave de la sesión* para crear un bloque pseudoaleatorio grande (*pad*) que se aplica en or-exclusiva al texto en claro para generar el texto cifrado. Este método tiene la ventaja de que el *pad* puede ser generado independientemente del texto en claro, lo que incrementa la velocidad de cifrado y descifrado.
- **CFB (Cipher Feedback Mode).** Variante del método anterior para mensajes muy largos.

Los métodos de encriptación históricamente son divididos en dos categorías: los *cifrados por sustitución* y los *cifrados por transposición*.

2.1.1 Cifrados por sustitución

Reemplazan símbolos o grupo de símbolos preferentemente, por otros símbolos o grupos de ellos pertenecientes al mismo alfabeto. Los métodos de sustitución se pueden clasificar en:

2.1.1.1 Sustitución simple o monoalfabéticos

Definición: Sea *A* un alfabeto de *q* símbolos y *M* un conjunto de todos los string de longitud *t* sobre *A*. Sea *P* el conjunto de todas las permutaciones sobre el conjunto *A*. Se define para cada $e \in P$ una transformación de encriptación E_e como:

$$E_e(m) = (e(m_1)e(m_2)...e(m_t)) = (c_1c_2...c_t) = c$$

donde $m = (m_1m_2...mt) \in M$. En otras palabras, para cada símbolo en la *t*-túpla se reemplaza (sustituye) por otro símbolo de *A* acorde a alguna permutación fija *e*. Para desencriptar $c = (c_1c_2...c_t)$ se utiliza la permutación inversa $d = e^{-1}$ y

$$D_d(c) = (d(c_1)d(c_2)...d(c_t) = (m_1m_2...m_t) = m.$$

El número de sustituciones distintas es q! y es independiente del tamaño del bloque de cifrado.

Ejemplo 2.1: Sea $\mathbf{Z_q}$ el conjunto con los siguientes elementos {0,1, ... , q – 1}. Sea M = C = $\mathbf{Z_q}$. Para $0 \leq K \leq q - 1$, se tiene que

$$E_K(x) = (x + K) \bmod q$$
$$D_K(y) = (y – K) \bmod q$$

donde $x \in M$ y $y \in C$.

☞ Para el caso particular en que la clave K = 3, el criptosistema a menudo es llamado *cifrado César*, porque supuestamente fue usado por Julio César.

Supongamos que trabajamos con el alfabeto en inglés que consta de 26 caracteres (q = 26) y nuestra clave K es 15.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Se tiene el siguiente mensaje: **controlcombinado**

Lo primero que hacemos es convertir el texto plano (mensaje) en una secuencia de enteros, obteniendo lo siguiente:

2	14	13	19	17	14	11	2
14	12	1	8	13	0	3	14

Luego se le suma 15 (clave K) a todos los elementos.

17	29	28	34	32	29	26	17
29	27	16	23	28	15	18	29

Posteriormente se realiza mod 26 a todos los elementos

17	3	2	8	6	3	0	17
3	1	16	23	2	15	18	3

Finalmente convertimos la secuencia de enteros a caracteres del alfabeto, obteniendo el siguiente criptograma: **rdcigdardbqxcpsd**

Para desencriptar el criptograma, se deberá convertir a secuencia de enteros, luego restarle 15 a cada valor. Posteriormente se obtiene el residuo de la división por 26 (función mod) y finalmente se convierte la secuencia de enteros obtenida en caracteres del alfabeto. Siguiente con el mismo ejemplo:

17	3	2	8	6	3	0	17
3	1	16	23	2	15	18	3

Restamos 15 y obtenemos el residuo de la división por 26. En el caso de que algún valor diera negativo se le suma 26 para obtener otro valor que pertenezca a $\mathbf{Z_{26}}$

2	14	13	19	17	14	11	2
14	12	1	8	13	0	3	14

Finalmente convertimos la secuencia de enteros obtenida a caracteres del alfabeto, obteniendo el mensaje o texto plano original: **controlcombinado**

Si analizamos el ejemplo anterior, podemos darnos cuenta que el cifrado por alteración no es seguro, ya que puede ser criptoanalizado, por ejemplo, con un método de fuerza bruta. Dado que el espacio de claves posibles (componente *K* del criptosistema) tiene solamente 26 elementos. Entonces es fácil probar todas las posibles claves de encriptación hasta encontrar alguna que devuelva un mensaje con significado o sentido para nosotros. Por ejemplo:

Dado el siguiente criptograma: **gvmtxskvejme**

Entonces lo que hacemos es probar todas las claves posibles:

Clave	Texto plano obtenido
K = 0	gvmtxskvejme
K = 1	fulswrjudild
K = 2	etkrvqitchkc
K = 3	dsjquphsbgjb
K = 4	**criptografia**

En este punto podemos determinar el texto plano y por lo tanto detener la búsqueda ya que la clave es K = 4. En promedio, un texto plano será encontrado luego de probar 26/2 = 13 claves de encriptación.

2.1.1.2 Homofónicos

Definición: Para cada símbolo $a \in A$, se asocia un conjunto *H(a)* de string de *t* símbolos, con la restricción que los conjuntos *H(a)*, $a \in A$ sean pares disjuntos. Una

sustitución homofónica reemplaza cada símbolo *a* en el bloque de texto plano con un string de *H(a)* elegido aleatoriamente. Para descriptar un string *c* de *t* símbolos, uno tiene que determinar un $a \in A$ tal que $c \in H(a)$. La clave para el cifrado consiste en los conjuntos *H(a)*.

Una definición más informal de un cifrado homofónico es que se utilizan múltiples sustituciones de símbolos para cada carácter, donde dicho número de sustituciones es proporcional a la frecuencia de aparición del carácter en el mensaje.

Ejemplo 2.2: El carácter "a" tiene una frecuencia promedio de aproximadamente del 8% en los textos en ingles. Entonces ocho símbolos pueden ser usados para representar a dicho carácter y dichas posibilidades se utilizaran de forma aleatoria en el criptograma. En principio se puede diseñar una sustitución homofónica de manera tal que ningún símbolo en el criptograma tenga frecuencia mayor al 1%, evitando así el análisis por frecuencia. A continuación se muestra un posible esquema de sustituciones:

A_i	$H(A_i)$
a	12 29 25 43 71 80 89 95
b	05 92
c	19 37 36
d	23 41 61 66
e	16 30 47 59 72 83 90 60 69 88 99 00
f	17 49
g	02 31
h	04 45 55 63 76 82
i	15 34 56 97 77 86
j	03
k	11
l	24 38 48 64
m	65 46
n	26 42 53 70 73 98
o	10 44 50 94 78 85 91
p	06 39
q	52
r	21 35 54 20 74 87
s	01 40 57 68 79 81
t	13 28 51 67 75 84 33 27 22
u	08 62 58
v	07
w	18 32
x	96
y	09 93
z	14

Donde:

- I es la cantidad de símbolos que tiene el alfabeto. En este caso: $0 \leq i \leq 25$
- A_i representa el i-ésimo símbolo del alfabeto
- $H(A_i)$ son las sustituciones posibles para el i-ésimo símbolo del alfabeto.

Ejemplo 2.3: Supongamos que *A* = {a,b}, H(a) = {00,10}, y H(b) = {01,11}. El bloque de mensaje **ab** se puede encriptar de la siguiente manera: 0001,0011,1001,1011. Observamos que el codominio de la función de encriptación (para los mensajes de longitud dos) esta formado por los siguientes pares disjuntos:

aa → {0000,0010,1000,1010}
ab → {0001,0011,1001,1011}
ba → {0100,0110,1100,1110}
bb → {0101,0111,1101,1111}

Normalmente los símbolos no ocurren con igual frecuencia en un texto plano. Con una simple sustitución esta propiedad es reflejada en el criptograma. Un cifrado homofónico puede ser usado para hacer que las frecuencias de ocurrencia en el criptograma sean uniformes y así evitar los ataques que estudian las frecuencias de aparición de los símbolos.

Mientras se tiende a ocultar la información de la frecuencia, como se menciono anteriormente, los patrones que las letras forman en el texto dan posibilidad para quebrarlo. Por ejemplo, en ingles la letra "q" en la mayoría de los casos es seguida por una "u". Dado que la "q" rara vez es utilizada en textos en ingles es por ello que se refleja en la tabla anterior con una sola sustitución posible. A diferencia de la "u" que tiene tres símbolos de sustitución posibles. De esta manera el mensaje encriptado va a tener distintos pares de símbolos, con un símbolo específico seguido de uno de los otros tres posibles y no otros. Lo que se esta haciendo con éste análisis es buscar reglas de asociación entre caracteres.

2.1.1.3 Polialfabéticos

Definición: Un cifrado de sustitución polialfabético es un bloque de cifrado de longitud *t* sobre un alfabeto *A* que tiene las siguientes propiedades:

- El espacio de clave *K* consiste en todos los conjuntos ordenados de *t* permutaciones $(p_1,p_2,\ldots,p_t)$, donde cada permutación p_i es definida en el conjunto *A.*
- La encriptación de un mensaje $m = (m_1m_2\ldots m_t)$ bajo la clave $e = (p_1,p_2,\ldots,p_t)$ es dado por $E_e(m) = (p_1(m_1)p_2(m_2)\ldots pt(m_t))$.
- La clave de desencriptación asociada con $e = (p_1,p_2,\ldots,p_t)$ es $d = (p_1^{-1},p_2^{-1},\ldots,p_t^{-1})$

Una definición más informal de un cifrado polialfabético es que la sustitución aplicada a cada carácter varía en función de la posición que ocupe éste dentro del mensaje. A diferencia de una sustitución homofónica, no se tiene un conjunto o secuencia fija de sustituciones posibles para cada carácter.

Ejemplo 2.4: Un método de éste tipo es el *cifrado de Vigénere* que debe su nombre a Blaise de Vigénere su creador y data del siglo XVI. Sea el alfabeto A con una cantidad de símbolos *q*, la clave K = (k1k2…kt) de longitud *t* y el mensaje que se desea encriptar m = (m1m2…mt). Entonces el método consiste:

$$E_K(m) = (\phi_1\phi_2\ldots\phi_t) = c$$
$$D_K(c) = (\phi_1^{-1}\phi_2^{-1}\ldots\phi_t^{-1}) = m$$

donde $\phi i = (m_i + K_i) \mod q \qquad 1 \le i \le t$
$\phi i^{-1} = (\phi_i - K_i) \mod q$

Supongamos que tenemos el alfabeto A = {A,B,…,Z} con *q* = 26 (longitud del alfabeto en inglés), nuestra clave es *datos* y el mensaje a encriptar es: *demostraciondelteorema*.

datos → K = (3,0,19,14,18) t = 5 (cantidad de símbolos que tiene la clave)

D	E	M	O	S	T	R	A	C	I	O
3	4	12	14	18	19	17	0	2	8	14
3	0	19	14	18	3	0	19	14	18	3
6	4	5	2	10	22	17	19	16	0	17

N	D	E	L	T	E	O	R	E	M	A
13	3	4	11	19	4	14	17	4	12	0
0	19	14	18	3	0	19	14	18	3	0
13	22	18	3	22	4	7	5	22	15	0

Entonces el criptograma es: **gefckwrtqarnwsdwehfwpa**

Para desencriptar el criptograma se debe utilizar la función D_K como se muestra:

6	4	5	2	10	22	17	19	16	0	17
3	0	19	14	18	3	0	19	14	18	3
3	4	12	14	18	19	17	0	2	8	14

13	22	18	3	22	4	7	5	22	15	0
0	19	14	18	3	0	19	14	18	3	0
13	3	4	11	19	4	14	17	4	12	0

2.1.1.4 Poligráficos

Todos los métodos de cifrado vistos hasta el momento son *monográficos*. Esta denominación se debe a que sustituyen un carácter por otro de una forma preestablecida. Dado que son vulnerables al análisis de frecuencia de aparición de los caracteres se han desarrollado esquemas basados en cifrar bloques de letras de una cierta longitud fija. Dichos métodos se denominan cifrados *poligráficos*. En otras palabras, en un cifrado por sustitución poligráfica se sustituyen un conjunto de caracteres juntos del mensaje por un grupo de caracteres (típicamente pares, dando lugar a un cifrado por diagramas).

Ejemplo 2.5: Cifrado de Hill: Fue inventado en 1929 por Lester S. Hill. Este método para realizar las sustituciones usa el álgebra lineal. Un bloque de t caracteres es considerado como un vector de t dimensiones, y es multiplicado por una matriz de dimensión txt. Vale destacar que todas las operaciones se realizan en modulo 26, ya que se considera que el alfabeto utilizado consta de esa cantidad de símbolos. Las componentes de la matriz es la clave de cifrado. Entonces para la clave K y el número entero fijo t, se define:

$E_K (m) = mK = c$
$D_K (c) = cK^{-1} = m$

donde:

- m = mensaje plano. Es un vector de t componentes
- c = criptograma. Es un vector de t componentes
- K = matriz inversible de dimensión txt.
- K^{-1} = matriz inversa de K en módulo 26.
- Todas las operaciones se deben realizar en módulo 26.
- Una condición suficiente para que K tenga inversa en módulo 26, es que el determinante de la matriz K mod 26 sea igual a 1, es decir, Det(K) mod 26 = 1.

Supongamos que deseamos cifrar el mensaje *organizacion* para ello utilizamos la siguiente clave:

O	R	G	A	N	I	Z	A	C	I	O	N
14	17	6	0	13	8	25	0	2	8	14	13

$K = \begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix}$, $\text{Det } K = 79 \rightarrow 79 \bmod 26 = 1$. Cumple la condición suficiente y entonces se puede calcular la inversa de la matriz de la siguiente manera: $K^{-1} = \begin{pmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{pmatrix}$

La matriz inversa es: $K^{-1} = \begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix}$

Encriptación:

Dado que t=2 se irán tomando de a dos caracteres del mensaje plano y multiplicando con la matriz K:

$$\text{paso 1}: c_1 = (14 \quad 17).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (21 \quad 16)$$

$$\text{paso 2}: c_2 = (6 \quad 0).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (24 \quad 6)$$

$$\text{paso 3}: c_3 = (13 \quad 8).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (8 \quad 17)$$

$$\text{paso 4}: c_4 = (25 \quad 0).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (22 \quad 25)$$

$$\text{paso 5}: c_5 = (2 \quad 8).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (16 \quad 6)$$

$$\text{paso 6}: c_6 = (14 \quad 13).\begin{pmatrix} 4 & 1 \\ 1 & 20 \end{pmatrix} = (17 \quad 14)$$

El criptograma obtenido es: **vqygirmzqgro**

Desencriptación:

$$\text{paso 1: } m_1 = (21 \quad 16).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (14 \quad 17)$$

$$\text{paso 2: } m_2 = (24 \quad 6).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (6 \quad 0)$$

$$\text{paso 3: } m_3 = (8 \quad 17).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (13 \quad 8)$$

$$\text{paso 4: } m_4 = (22 \quad 25).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (25 \quad 0)$$

$$\text{paso 5: } m_5 = (16 \quad 6).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (2 \quad 8)$$

$$\text{paso 6: } m_6 = (17 \quad 14).\begin{pmatrix} 20 & -1 \\ -1 & 4 \end{pmatrix} = (14 \quad 13)$$

Con lo cual se obtiene el mensaje original.

Ejemplo 2.6: PlayFair: inventado en 1854 por Charles Wheatstone para comunicaciones telegráficas secretas. Fue utilizado por el Reino Unido en la Primera Guerra Mundial. El método consiste en separar el texto plano en diagramas y proceder a su cifrado de acuerdo a una matriz alfabética de dimensiones 5 x 5, en la cual se encuentran representadas las 26 letras del alfabeto ingles. Dado que la que esta sobrando un símbolo del alfabeto se considera a al carácter "i" y "j" como uno solo. Como se muestra a continuación:

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

La clave utilizada en éste método es el orden que se le da a los caracteres en la matriz anterior, ya sea arbitrario o aleatorio. Una posibilidad para determinar ese orden puede ser utilizar una frase quitándole los caracteres repetidos y a continuación el resto de los caracteres del alfabeto que no contiene dicha frase.

Para cifrar es necesario seguir las siguientes reglas:

1. Si los símbolos en el diagrama son iguales, se reemplaza el segundo símbolo por otro acordado con anterioridad (por lo general es "X"). Se encripta el nuevo par.
2. Si los caracteres aparecen en la misma fila de la matriz, para cada carácter del par la sustitución se realiza con el carácter ubicado inmediatamente a la derecha. En el caso que se termine la fila se continúa para abajo.
3. Si los caracteres aparecen en la misma columna de la matriz, para cada carácter del par la sustitución se realiza con el carácter ubicado inmediatamente abajo. Si se termina la columna se continúa hacia la derecha.
4. Si los caracteres aparecen en distinta fila y columna, se arma un rectángulo en la matriz que contiene a dichos caracteres como vértices. La sustitución para cada carácter del par se realiza tomando, de los dos vértices restantes, el que esta ubicado en la misma fila.

Supongamos que deseamos cifrar el mensaje *organizacion* y que la matriz tiene el siguiente ordenamiento de símbolos:

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

Encriptación:

Primero separamos el mensaje plano en pares o diagramas: OR GA NI ZA CI ON

Paso Diagrama Regla Cifrado Matriz

Paso	Diagrama	Matriz	Regla	Sustitución	Cifrado
1	OR	D P **F** U T H C **R** Q K W Y X **O** E G I/J M N B A L Z S V	4	R → Q O → X	**XQ**
2	GA	**D** P F U T **H** C R Q K **W** Y X O E **G** I/J M N B **A** L Z S V	3	G → A A → P	**AP**
3	NI	D P F U T H C R Q K W Y X O E G **I/J** M **N** B A L Z S V	2	N → B I → M	**BM**
4	ZA	D P F U T H C R Q K W Y X O E G I/J M N B **A** L **Z** S V	2	Z → S A → L	**SL**
5	CI	D **P** F U T H **C** R Q K W Y X O E G **I/J** M N B A L Z S V	3	C → Y I → L	**YL**
6	ON	D P F **U** T H C R **Q** K W Y X **O** E G I/J M **N** B A L Z **S** V	3	O → N N → S	**NS**

Entonces el criptograma obtenido es: **XQAPBMSLYLNS**

2.1.2 Cifrados por transposición

Este tipo de mecanismo de cifrado no sustituye unos símbolos por otros, sino que cambia su orden (permutación) dentro del texto alterando la aparición estadística. También es llamado cifrado por permutación.

Definición: Considerar un esquema de encriptación por bloque con clave simétrica con una longitud de bloque *t*. Sea *K* el conjunto de todas las permutaciones del conjunto {1,2,…,t}. Para cada $e \in K$ se define la función de encriptación:

$E_e(m) = (m_{e(1)}m_{e(2)}...me_{(t)})$

Donde $m = (m_1m_2...m_t) \in M$, el espacio de mensajes. El conjunto de todas esas transformaciones es denominado *cifrado simple por transposición*. La clave de desencriptación para *e* es la permutación inversa $d = e^{-1}$. Para desencriptar $c = (c_1c_2...c_t)$ se utiliza la función

$D_d(c) = (c_{d(1)}c_{d(2)}...c_{d(t)})$

Un cifrado simple por transposición mantiene el número de símbolos de un tipo dado en el bloque, y por ello es fácil de criptoanalizar.

Ejemplo 2.7: Sea t un número entero fijo positivo. La componente K del criptosistema esta formada por t! elementos, que son todas permutaciones posibles. Por ejemplo para la clave p_k:

$E_{pk}(m_1m_2...m_t) = (m_{pk(1)}m_{pk(2)}...m_{pk(t)}) = c$
$D_{pk}^{-1}(m_{pk(1)}m_{pk(2)}...m_{pk(t)}) = (m_1m_2...m_t) = m$

Donde p_k^{-1} es la inversa de la permutación p_k.

Supongamos que t=6, m=establecerelprecio y que muestra clave pk es la siguiente:

P_K	
M	**C**
1	4
2	1
3	3
4	6
5	2
6	5

P_K^{-1}	
C	**M**
1	2
2	5
3	3
4	1
5	6
6	4

La tabla P_K se lee de la siguiente manera: el carácter en la posición 1 del mensaje plano pasará a ocupar la posición 4 en el criptograma. Ésta se utiliza para cifrar el mensaje. En cambio la tabla P_K^{-1} se utiliza para descifrar el mensaje y lee de forma análoga a la tabla P_K, con la diferencia de se parte del criptograma y no del mensaje.

Encriptación:

Se divide el mensaje en bloques de tamaño 6, dado que t=6 (longitud de la clave) y se aplica la permutación P_K:

Paso	Mensaje	Cifrado
1	establ	sbtela
2	ecerel	ceeelr
3	Precio	riepoc

Entonces el criptograma resulta: **sbtelaceeelrriepoc**

Desencriptación:

Se divide el criptograma en bloques de tamaño 6 y se aplica la permutación P_K^{-1}

Paso	Cifrado	Mensaje
1	sbtela	establ
2	ceeelr	ecerel
3	riepoc	Precio

Entonces el mensaje plano es el original: **establecerelprecio**

Si se analiza el método anterior se puede llegar a la conclusión de que es un caso particular del método de Hill, en donde la matriz K de dimensión txt se obtiene al escribir la tabla P_K en forma matricial. Para el ejemplo anterior la matriz K es la siguiente:

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

Si se transpone la matriz K se obtiene la matriz inversa K^{-1}

$$K^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Ejemplo 2.8: Otro ejemplo de éste tipo de cifrado es el *transposición por columna*. La clave de cifrado puede ser una palabra o frase con la única restricción de que no se repita ningún carácter en ella. Supongamos que tenemos lo siguiente:

Clave = segunda
Mensaje = establecenlassiguientescaracteristicasimportantes

El propósito de la clave es para enumerar las columnas, donde la columna 1 estará dada por el menor de los caracteres de la clave (desde el comienzo del alfabeto). El mensaje es escrito de forma horizontal, en filas, en la matriz. El criptograma es leído por columnas, comenzando con la columna que tiene el carácter de clave menor.

s	e	g	u	n	d	a
6	2	4	7	5	3	1
e	s	t	a	b	l	e
c	e	n	l	a	s	s
i	g	u	i	e	n	t
e	s	c	a	r	a	c
t	e	r	i	s	t	i
c	a	s	i	m	p	o
r	t	a	n	t	e	s

Entonces el criptograma es:

estciossegseatlsnatpetnucrsabaersmtecietcraliaiin

2.1.3 Cifrado de producto

El cifrado por sustitución y por transposición individualmente no proveen un grado muy alto de seguridad. Sin embargo, si se combinan éstas transformaciones es posible obtener un método de cifrado más seguro que se conoce como *cifrado de producto*. Estas técnicas consisten básicamente en dividir el mensaje en bloque de tamaño fijo y aplicar la función de cifrado a cada uno de ellos. Un ejemplo de un cifrado por producto es la composición de t transformaciones $E_{k1}E_{k2}...E_{kt}$ donde $t \geq 2$ y cada E_{ki} $1 \leq i \leq t$, es una transformación de sustitución o transposición.

La mayoría de los algoritmos se basan en diferentes capas de sustituciones y permutaciones. Donde en muchos casos el criptosistema no es más que una operación combinada de dichas transformaciones, repetida n veces, como ocurre con el algoritmo DES, que se verá en el próximo capítulo.

Ejemplo 2.9: Supongamos que tenemos el mensaje plano *demostraciondelteorema* y que utilizamos las siguientes transformaciones:

$E_K^{(1)}$ = { Cifrado de Vigénere con clave = datos }

$E_{P_K}^{(2)}$ = { Cifrado por transposición con P_K igual a la utiliza en el Ejemplo 2.7 }

Entonces lo que tenemos que hacer es: $E_{P_K}(E_K(m))$

$E_K(m)$ = **gefckwrtqarnwsdwehfwpa** (se resolvió en el ejemplo 2.4)

Entonces si aplicamos la permutación P_K a $E_K(m)$ se obtiene:

$E_{Pk}(E_K(m))$ = **ekfgwctrqrnasedwhww*pf*a**

Se tuvo que completar el mensaje con un carácter especial (asterisco), ya que la longitud del mensaje no era multiplico de 6.

Una cuestión a tener en cuenta en los cifrados por producto es la posibilidad de que posean *estructura de grupo*. Se dice que un cifrado tiene *estructura de grupo* si se cumple la siguiente propiedad:

$$\forall k_1, k_2 \quad \exists k_3 \text{ tal que } \quad E_{k_2}(E_{k_1}(m)) = E_{k_3}(m)$$

esto es, si hacemos dos cifrados encadenados con k_1 y k_2, existe una clave k_3 que realiza la transformación equivalente.

Entonces lo que se busca es que el algoritmo criptográfico carezca de este tipo de estructura, ya que si ciframos un mensaje primero con la clave k_1 y el resultado con la clave k_2, es como si hubiéramos empleado una clave de longitud doble, aumentando la seguridad del sistema. Si, por el contrario, la transformación criptográfica presentara estructura de grupo, esto hubiera sido equivalente a cifrar

el mensaje una única vez con una tercera clave, con lo que no habríamos ganado nada.

2.2 Cifrado de Flujo

También llamado cifrado por secuencia, consiste en emplear una secuencia aleatoria de igual longitud que el mensaje, a diferencia de los cifrados por bloques donde cada bloque estaba determinado por la longitud de la clave. La idea consiste en utilizar dicha cadena o secuencia aleatoria una única vez, combinándola mediante alguna función simple, y por supuesto biyectiva, con el mensaje plano carácter a carácter. Claramente éste método presenta el grave inconveniente de que la clave es tan larga como el mensaje, entonces carece de utilidad práctica en la mayoría de los casos. Pero si disponemos de un generador pseudoaleatorio capaz de generar secuencia *criptográficamente aleatorias*, de forma que la longitud de los posibles ciclos sea extremadamente grande (estaríamos garantizando el usar una secuencia dada una única vez), se podría emplear la semilla del generador como clave. Entonces todo aquel que conozca la semilla podrá reconstruir la secuencia pseudoaleatoria y de esta forma descifrar el mensaje.

En ésta sección no se mostraran algoritmos que utilizan la idea del generador pseudoaleatorio, dado que exceden el objetivo de la materia. Pero si el lector esta interesado en profundizar el tema puede seguir su lectura en [Robshaw, 1995b].

2.2.1 One Time Pad

Dicho criptosistema consiste en emplear una secuencia aleatoria de igual longitud que el mensaje que se desea encriptar, que se utilizaría por una única vez. Entonces dicha secuencia aleatoria y el texto plano se combinan mediante alguna función simple y reversible carácter a carácter. Un punto débil de éste criptosistema es que la clave no puede ser memorizada, entonces tanto el emisor como el receptor deben cargar una copia de ella con ellos.

A continuación se muestra un ejemplo de cómo funciona dicho método. La función utilizada para la combinación del texto plano con la cadena aleatoria es la XOR.

Supongamos que nuestro texto plano o mensaje a transmitir es: **criptografia**

Lo primero que hacemos es convertir dicho mensaje en una secuencia de bits, para ello utilizamos la representación ASCII (8-bits) de cada carácter.

	c	r	i	p	t	o
Mensaje	01100011	01110010	01101001	01110000	01110100	01101111
S.Aleatoria	01011010	01110001	10100000	00101000	00010010	10101110
Criptograma	00111001	00000011	11001001	01011000	01100110	11000001

	g	r	a	f	i	a
Mensaje	01100111	01110010	01100001	01100110	01101001	01100001
S.Aleatoria	10101001	11110000	01111111	11111111	11100101	00100011
Criptograma	11001110	10000010	00011110	10011001	10001100	01000010

Entonces lo que obtenemos es lo siguiente:

Criptograma (ascii): **9♥╔Xf╩╬éAÖîB**
Clave (ascii): **Zqá(↕«®-⌂ Õ#**

Claramente se puede ver en el ejemplo anterior que es imposible memorizar la clave aleatoria.

Si dicho método se quiere criptoanalizar, por ejemplo, por fuerza bruta es imposible obtener el texto plano original ya que al ir probando con todas las posibles cadenas aleatorias utilizadas en el proceso de encriptación para el ejemplo anterior se podría dar lo siguiente:

Criptograma	00111001	00000011	11001001	01011000	01100110	11000001
S.Aleatoria	01001001	01110001	10100110	00111111	00010100	10100000
Mensaje	01110000	01110010	01101111	01100111	01110010	01100001

Criptograma	11001110	10000010	00011110	10011001	10001100	01000010
S.Aleatoria	10100011	01100011	01110001	11110000	01100011	00101100
Mensaje	01101101	01100001	01100011	01101001	01101111	01101110

Entonces el criptograma obtenido es: **programación**. El criptoanalista detendría su búsqueda ya que el texto plano obtenido tiene “sentido” para él. Pero éste no coincide con el texto plano original.

En resumen, el algoritmo anterior es incondicionalmente seguro, aunque, como las claves son del mismo tamaño que la entrada, es de poca utilidad práctica.

3 Criptografía de clave privada

La criptografía moderna utiliza las mismas ideas que la criptografía clásica: transposiciones o sustituciones. Tradicionalmente los criptógrafos han usado algoritmos simples. Pero actualmente, el objetivo es hacer que los algoritmos de encriptación sean tan complejos que el criptoanalista, teniendo en su poder criptogramas, no pueda obtener los mensajes planos correspondientes sin conocer la clave de encriptación.

En la figura 3.1 se muestra un modelo de encriptación para el cifrado de clave privada o simétrico.

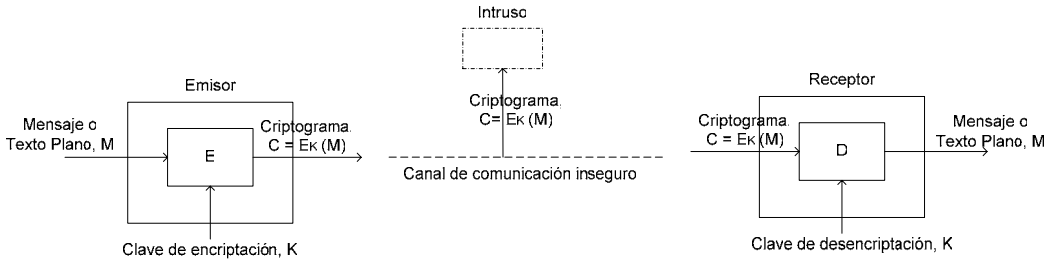

Figura 3.1 - Modelo de encriptación – Clave privada o simétrica.

El Emisor desea enviarle al receptor un mensaje. Para ello dicho mensaje que se desea transmitir es transformado por una función o método de encriptación (E) que tiene como parámetro la clave K. La salida del proceso de encriptación (criptograma) es transmitido por un canal de comunicación. Asumimos que el intruso o enemigo escucha en el canal de comunicación y que puede hacer copias fieles de los criptogramas. Sin embargo, dicha parte no conoce la clave de desencriptación y por lo tanto no podrá desencriptar fácilmente dicho criptograma. Algunas veces el intruso no solo pude escuchar en el medio de comunicación sino también enviar mensajes. Con lo cual puede enviar sus propios mensajes o modificar legítimos mensajes antes de que éstos lleguen al receptor deseado. Del otro lado del canal de comunicación se encuentra el receptor, quién le aplica la transformación inversa o método de desencriptación (D), que tiene como parámetro la clave K, al criptograma recibido. De ésta manera el receptor recupera el mensaje plano.

3.1 DES

El DES (*Data Encryption Standard*) es desde 1977 de uso obligatorio en el cifrado de informaciones gubernamentales no clasificadas (anunciado por el *National Bureau of Standards*, USA). Este criptosistema fue desarrollado por IBM como una variación de un criptosistema anterior, Lucifer, y posteriormente, tras algunas comprobaciones llevadas a cabo por la NSA estadounidense, pasó a transformarse en el que hoy conocemos como DES. DES puede ser implementado tanto en *software* como en chips con tecnología VLSI (*Very Large Scale Integration*), alcanzando en *hardware* una velocidad de hasta 50 Mbs. Un ejemplo de

implementación por *hard* puede ser PC-Encryptor, de Eracom, y un ejemplo de implementación por *software* es DES-LOCK, de la empresa Oceanics.

DES es un sistema de clave privada tanto de cifrado como de descifrado; posee una clave de entrada con una longitud de 64 bits, produciendo una salida también de 64 bits, con una clave de 56 *bits* (el octavo bit de cada byte es de paridad), llamada clave externa, en la que reside toda la seguridad del criptosistema, ya que el algoritmo es de dominio público. Cada trozo de 64 bits de los datos se desordena según un esquema fijo a partir de una permutación inicial conocida como IP. A continuación, se divide cada uno de los trozos en dos mitades de 32 bits, que se someten a un algoritmo durante 16 iteraciones. Este algoritmo básico que se repite 16 veces (llamadas vueltas), utiliza en cada una de ellas 48 de los 56 bits de la clave (estos 48 bits se denominan clave interna, diferente en cada vuelta); las claves internas se utilizan en un orden para cifrar texto y en el orden inverso para descifrarlo. En cada una de las vueltas se realizan permutaciones, sustituciones no lineales (que constituyen en sí el núcleo del algoritmo DES) y operaciones lógicas básicas, como la XOR. La mitad derecha se transfiere a la mitad izquierda sin ningún cambio; también se expande de 32 hasta 48 bits, utilizando para ello una simple duplicación. El resultado final de una iteración es un XOR con la clave interna de la vuelta correspondiente, y esta salida se divide en bloques de 6 bits, cada uno de los cuales se somete a una sustitución en un bloque de 4 bits (bloque-S, con un rango 0...63) dando una salida también de 4 bits (rango decimal 0...15) que a su vez se recombina con una permutación en un registro con longitud 32 bits. Con el contenido de este registro se efectúa una operación XOR sobre la mitad izquierda de los datos originales, convirtiéndose el nuevo resultado en una salida (parte derecha) de 32 bits; transcurridas las dieciséis vueltas, las dos mitades finales (de 32 bits cada una) se recombinan con una permutación contraria a la realizada al principio (IP), y el resultado es un criptograma de 64 bits.

Aunque no ha sido posible demostrar rigurosamente la debilidad del criptosistema DES, y actualmente es uno de los más utilizados en el mundo entero, parece claro que con las actuales computadoras y su elevada potencia de cálculo una clave de 56 bits (en la que recordemos, reside toda la seguridad del DES) es fácilmente vulnerable frente a un ataque exhaustivo o por fuerza bruta en el que se prueben combinaciones de esos 56 bits. Hay que resaltar que el tamaño inicial de la clave, en el diseño de IBM, era de 128 bits; la razón de la disminución no se ha hecho pública hasta el momento. Sin embargo, mucha gente sospecha que dicha reducción de la clave se hizo para que la NSA pudiera romperlo y no así pequeñas organizaciones que no tuvieran el equipamiento que dicha agencia.

En 1977, dos investigadores de la criptografía de Stanford, Diffie y Hellman, diseñaron una máquina para romper el DES y estimaban que podría ser construido por 20 millones de dólares. Dado un pedazo pequeño de texto plano y de texto cifrado, esta máquina podía encontrar la llave por la búsqueda exhaustiva o por fuerza bruta del espacio dominante 2^{56} entradas en un plazo menor a un día. Hoy en día, tal máquina costaría aproximadamente un millón de dólares.

A pesar de dicho problema, DES sigue siendo ampliamente utilizado en una 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 que emplea una clave demasiado corta, como se menciono anteriormente, lo cual hace que con el avance actual de las computadoras los ataques por fuerza bruta comiencen a ser opciones realistas. Mucha gente se resiste a abandonar este

algoritmo, precisamente porque ha sido capaz de sobrevivir durante veinte años sin mostrar ninguna debilidad en su diseño, y prefieren proponer variantes para evitar el riesgo de tener que confiar en nuevos algoritmos y por otro lado aprovechar gran parte de las implementaciones por hardware existentes de DES.

En la figura 3.2 se muestra un esquema del algoritmo de DES. Si el lector esta interesado en profundizar más el tema puede ser en [Schneier, 1995].

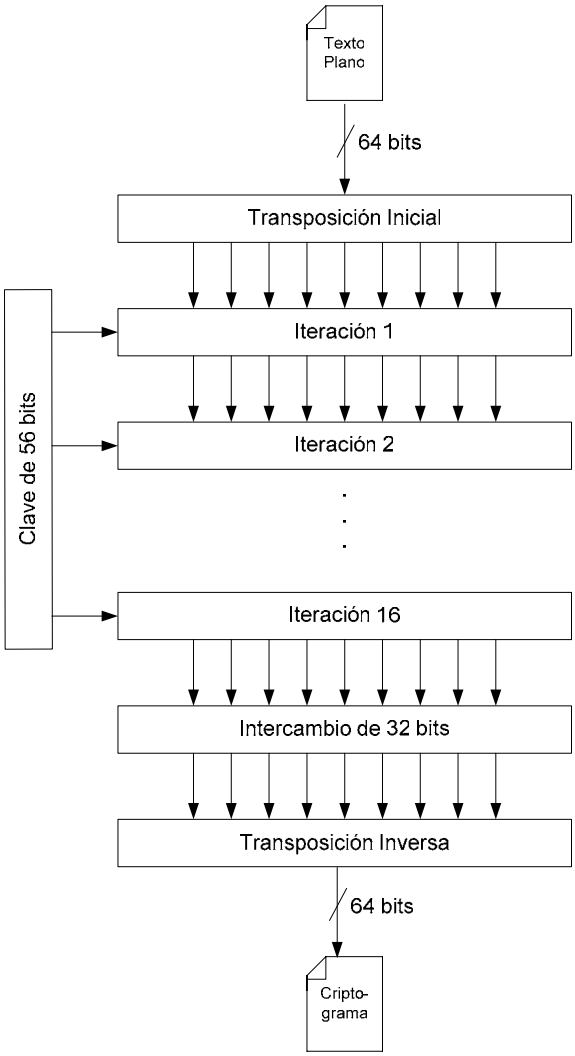

Figura 3.2 – Esquema algoritmo DES

3.2 Triple DES

En 1979, IBM se dio cuenta que la longitud dominante del DES era demasiado corta e ideó una manera de aumentarla con eficacia, usando el cifrado triple (Tuchman, 1979). El método elegido, que se ha incorporado desde entonces en el estándar internacional 8732, se ilustra en la Figura 3.3. Aquí se utilizan dos claves y tres etapas. En la primera etapa, se cifra el texto plano usando el DES en la manera generalmente con K_1. En la segunda etapa, el DES se ejecuta en modo de descifrado, usando K_2 como el clave. Finalmente, otro cifrado del DES se hace con K_1.

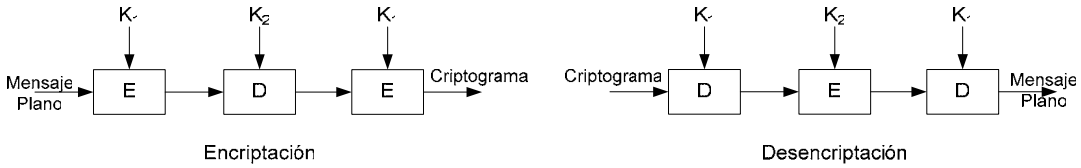

Figura 3.3– Esquema algoritmo TDES

Este diseño da lugar inmediatamente a dos preguntas. ¿Primero, por qué solamente dos claves se utilizan, en vez de tres? ¿En segundo lugar, por qué EDE (Encriptar Desencriptar Encriptar) se utiliza, en vez de EEE (Encriptar Encriptar Encriptar)? La razón que dos claves están utilizados es que incluso los criptógrafos más paranoicos creen que 112 dígitos binarios son adecuados para las aplicaciones comerciales rutinarias por el tiempo que duran. (entre criptógrafos, la paranoia se considera una característica, no un fallo de funcionamiento). Utilizar 168 dígitos binarios agregaría costos operativos innecesarios y además el transporte de otra clave, para lograr un pequeño aumento de seguridad.

La razón de cifrar, de desencriptar, y después de cifrar otra vez es compatibilidad hacia atrás con los sistemas existentes del DES de solo una clave. Las funciones del cifrado y del desciframiento son mapeos entre los conjuntos de números 64 bits. Desde un punto de vista criptográfico, los dos mapeos son igualmente fuertes. Usando EDE, sin embargo, en vez de EEE, una computadora que usa el cifrado triple puede hablar con otra que utiliza la encriptación simple solamente usando $K_1 = K_2$. Esta característica permite que el cifrado triple sea puestos en fase adentro gradualmente, algo de ninguna preocupación a los criptógrafos académicos, pero de importancia considerable para IBM y a sus clientes.

3.3 AES

En Enero de 1997, el National Institute of Standards and Technology (NIST) inició un esfuerzo para definir un nuevo algoritmo estándar en sustitución de DES, el que se llamaría Advanced Encryption Standard (AES). El primer llamado formal por propuestas de algoritmos se realizó en Septiembre de 1997.

Las condiciones del algoritmo eran que fuese desclasificado, público y gratis a nivel mundial. Además debía ser un algoritmo simétrico que permitiera como mínimo encriptar bloques de 128 bits, usando llaves de 128, 192 y 256 bits.

En Agosto de 1998, luego de la primera conferencia para AES, NIST anunció que quince algoritmos habían sido preseleccionados. En Marzo de 1999 se realizó la segunda conferencia orientada a AES, de donde salieron cinco algoritmos finalistas: MARS [Burwick et al., 1999], RC6 [Rivest et al., 1998], Rijndael [Daemen & Rijmen, 1998], Serpent [Biham et al., 1998] y Twofish [Shneier et al., 1998].

Luego de una tercera conferencia y dos rondas de recepción de comentarios públicos, el 2 de Octubre del 2000, NIST anunció que el algoritmo seleccionado para proponer como AES era Rijndael.

Rijndael fue diseñado por Joan Daemen y Vincent Rijmen. El tamaño del bloque a encriptar como la llave son de largo variables, puede encriptar bloques de 128, 192 o 256 bits utilizando llaves de 128, 192 o 256 bits.

Rijndael, o mejor dicho AES, aún no es usado ampliamente producto de ser un estándar muy reciente.

3.4 Criptoanálisis de algoritmos simétricos

Se podría decir que el criptoanálisis se comenzó a estudiar seriamente con la aparición de DES. Mucha gente desconfiaba (y aún desconfía) del algoritmo propuesto por la NSA. Se dice que existen estructuras extrañas, que muchos consideran sencillamente puertas traseras colocadas por la Agencia para facilitarles el descifrado de los mensajes. Nadie ha podido aún demostrar ni desmentir este punto. Lo único cierto es que el interés por buscar posibles debilidades en él ha llevado a desarrollar técnicas que posteriormente han tenido éxito con otros algoritmos.

Ni que decir tiene que estos métodos no han conseguido doblegar a DES, pero si representan mecanismos significativamente más eficientes que la fuerza bruta para criptoanalizar un mensaje. Los dos métodos que vamos a comentar parten de que disponemos de grandes cantidades de pares texto plano-criptograma obtenidos con la clave que queremos descubrir.

3.4.1 Criptoanálisis diferencial

Esta técnica puede utilizarse para atacar cualquier cifrado en bloques. Se empieza con un par de bloques de texto plano que difieren sólo en una cantidad pequeña de bits y se va observando cuidadosamente lo que ocurre en cada iteración interna a medida que avanza la encriptación. En muchos casos, algunos patrones son mucho más comunes que otros, y esta observación conduce a un ataque probabilística.

3.4.2 Criptoanálisis lineal

Este método pude descifrar el DES con sólo 243 textos planos conocidos. Funciona aplicando un OR-exclusiva a ciertos bits del texto plano y el texto cifrado en conjunto y buscando patrones en el resultado. Al hacerse repetidamente, la mitad de los bits deben ser ceros y la otra mitad unos. Sin embargo, con frecuencia los cifrados introducen una desviación en una dirección o en la otra, y esta desviación, por pequeña que sea, puede explotarse para reducir el factor de trabajo.

4 Criptografía de clave pública

En un esquema de encriptación de clave pública o asimétrico cada usuario participante en la comunicación posee una clave pública K_{pu} y una clave privada K_{pr}. Donde la clave pública es conocida por todo el mundo, mientras que la clave privada es conocida únicamente por su dueño. La clave pública define una transformación de encriptación E_{Kpu}, mientras que la clave privada tiene asociada una transformación de desencriptación D_{Kpr} inversa a la primera.

Para que el esquema se considere seguro tanto la transformación E como D deben cumplir los siguientes tres requerimientos:

- Si se aplica la transformación D al criptograma, se recupera el mensaje plano original. Sino se cumple esta condición el usuario receptor no tendría posibilidad de desencriptar el criptograma. D(E(m)) = m.

- La tarea de calcular K_{pr} dado K_{pu} es computacionalmente inviable.

- La transformación de encriptación E no debe poder ser quebrada por un ataque de texto plano escogido. Si se cumple esta condición no existe otra razón para que dicha transformación no sea pública.

Cualquier usuario que desee enviar un mensaje *m* a otro, lo que tiene que hacer es conseguir una copia autentica de la clave pública del receptor y aplicarle la transformación E con dicha clave, es decir, $c = E_{Kpu}(m)$. Posteriormente el emisor envía el criptograma obtenido al receptor, él cual recuperará el mensaje plano aplicándole la transformación inversa al criptograma recibido, $m = D_{Kpr}(c)$.

En la figura 4.1 se muestra un modelo de encriptación de clave pública o asimétrico.

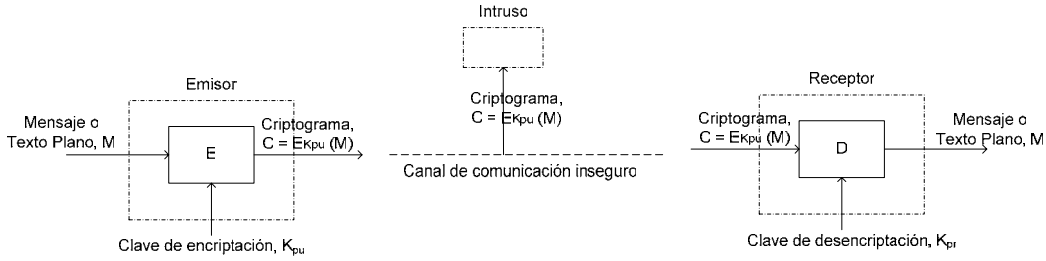

Figura 4.1 - Modelo de encriptación – Clave pública o asimétrica.

Una primera ventaja de éste esquema es que generalmente es más fácil garantizar la autenticidad de una clave pública que distribuir claves privadas como se requiere en un esquema simétrico.

El objetivo principal de la encriptación asimétrica es la de proporcionar confidencialidad. Dado que la transformación de encriptación E es de público conocimiento, la clave pública sola no puede proveer autentificación del origen ni integridad de los datos. Para lograr dichos objetivos necesitamos utilizar otras técnicas que se verán más adelante en este apunte.

Dicho criptosistema es típicamente más lento que algoritmos simétricos, como por ejemplo el DES. Por esta razón lo que se hace en la práctica es encriptar el mensaje a enviar como un algoritmo simétrico y enviar la clave utilizada encriptada con algún algoritmo asimétrico.

4.1 RSA

El criptosistema RSA [Rivest et al., 1978] es el criptosistema asimétrico más ampliamente utilizado. Puede ser utilizado para proveer tanto confidencialidad como en firma digital. La seguridad del criptosistema radica en la dificultad de la factorización de números enteros largos.

4.1.1 Generación de las claves

Cada usuario debe crear una clave pública RSA y su correspondiente clave privada. Los pasos para la creación de dichas claves se puede resumir de la siguiente manera:

1. Generar dos números largos aleatorios primos y distintos, p y q. Ambos deben poseer la misma cantidad de dígitos (típicamente 1024 bits).
2. Obtener n = p x q y $\phi(n) = (p-1) \times (q-1)$.
3. Seleccionar un número relativamente primo de $\phi(n)$ y denominarlo d. Se debe cumplir: gcd $(d,\phi(n)) = 1$. "gcd" significa "greatest common divisor". Entonces lo que se pretende es que d y $\phi(n)$ no tengan ningún divisor común.
4. Finalmente se calcula el entero e, con respecto a p,q y d que será el inverso multiplicativo de d, modulo $\phi(n)$. Se debe cumplir: $e \times d = 1 \pmod{\phi(n)}$.
5. La clave pública consiste en el par (e,n) y la privada en el par (d,n).

Los enteros e y d en la generación de la clave RSA se denominan el exponente de encriptación y el exponente de desencriptación, respectivamente. Mientras que n se denomina el modulo.

4.1.1.1 Como seleccionar el entero d

Es bastante fácil elegir un número d que es relativamente primo con $\phi(n)$. Por ejemplo, cualquier numero primo mayor que máx {p,q} puede ser. Es importante que d sea elegido de un conjunto amplio de manera tal que el criptoanalista no pueda encontrarlo con una búsqueda directa.

4.1.1.2 Como calcular el entero e dado d y $\phi(n)$

Se utiliza una variación del algoritmo de Euclides para calcular el máximo común divisor entre dos enteros [Shneier, 1996]. El algoritmo se muestra a continuación:

Entrada: Dos números no negativos a y b con $a \geq b$

Salida: d = gcd (a,b) y los enteros x e y que satisfacen ax + by = d. Lo cual por aritmética modular es equivalente a: by = d (mod –xa)

1. Si b = 0 $\Rightarrow$ d $\leftarrow$ a, x $\leftarrow$ 1, y $\leftarrow$ 0. Retornar (d,x,y)
2. $x_2 \leftarrow 1$, $x_1 \leftarrow 0$, $y_2 \leftarrow 0$, $y_1 \leftarrow 1$
3. Mientras b > 0 hacer
   3.1 q $\leftarrow$ Piso (a/b), r $\leftarrow$ a – qb, x $\leftarrow x_2 - qx_1$, y $\leftarrow y_2 - qy_1$
   3.2 a $\leftarrow$ b, b $\leftarrow$ r, $x_2 \leftarrow x_1$, $x_1 \leftarrow$ x, $y_2 \leftarrow y_1$, $y_1 \leftarrow$ y
4. d $\leftarrow$ a, x $\leftarrow$ x2, y $\leftarrow$ y2. Retornar (d,x,y)

Ejemplo 4.1: Supongamos que tenemos $\phi(n) = 2668$ y d = 157 y queremos calcular e utilizando el algoritmo anterior.

Entrada: a = 2668 y b = 157

Paso	q	r	x	y	a	b	x_2	x_1	y_2	y_1
-					2668	157	1	0	0	1
1	16	156	1	-16	157	156	0	1	1	-16
2	1	1	-1	17	156	1	1	-1	-16	17
3	0	0	1	-16	1	0	-1	1	17	-16

Salida: d = 1, x = -1, y = 17.

Entonces el número e es igual a 17.

4.1.2 Encriptación y Desencriptación

Para encriptar un mensaje m usando la clave pública (e,n) se procede de la siguiente manera:

- Representar el mensaje como enteros entre 0 y n - 1. Se debe partir el mensaje en una serie de bloques y representar cada bloque como un entero. Se puede utilizar cualquier representación estándar. El propósito de este paso no es encriptar el mensaje, sino obtener una representación numérica requerida para la encriptación.
- Se calcula $c = m^e \bmod n$. Entonces el criptograma es el resto de dividir m^e por n.

Para desencriptar se utiliza la clave privada (d,n) y se calcula $c^d \bmod n$. Posteriormente se deberá hacer la presentación numérica inversa utilizada antes de la encriptación.

Ejemplo 4.2: Supongamos que tenemos el mensaje plano *criptografia* y elegimos p=47, q=59, n=pq = 47 x 59 = 2773 y d = 157. Entonces $\phi(2773) = 46 \times 58 = 2668$. Según el ejemplo 4.1 obtuvimos que e = 17.

Encriptación:

Representación numérica a utilizar:

A	B	C	D	E	F	G	H	I	J	K	L	M
00	01	02	03	04	05	06	07	08	09	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Mensaje Plano: criptografia → 02170815191406170005080 0

Paso	Mensaje	Cálculo	Cifrado
1	0217	$0217^{17} \bmod 2773$	1219
2	0815	$0815^{17} \bmod 2773$	0635
3	1914	$1914^{17} \bmod 2773$	2575
4	0617	$0617^{17} \bmod 2773$	2375
5	0005	$0005^{17} \bmod 2773$	0508
6	0800	$0800^{17} \bmod 2773$	1505

Desencriptación:

Paso	Cifrado	Cálculo	Mensaje
1	1219	$1219^{157} \bmod 2773$	0217
2	0635	$0635^{157} \bmod 2773$	0815
3	2575	$2575^{157} \bmod 2773$	1914
4	2375	$2375^{157} \bmod 2773$	0617
5	0508	$0508^{157} \bmod 2773$	0005
6	1505	$1505^{157} \bmod 2773$	0800

En la práctica, RSA es utilizado generalmente para transportar la clave de encriptación de algún método de encriptación simétrico o para encriptar pequeña información.

4.1.3 Criptoanálisis

A continuación se mostraran maneras en las que un criptoanalista podría intentar determinar la clave privada de desencriptación a partir de la clave pública de encriptación. No se consideraran maneras de proteger la clave privada contra hurto.

4.1.3.1 Factorización de n

Si un criptoanalista enemigo logra descomponer en factores a n "rompio" el método. Dado que si obtiene los factores de n puede calcular el $\phi(n)$ y así el número d. Afortunadamente, descomponer en factores un número parece ser mucho más difícil que determinar si un número es primo o no.

Existen una gran cantidad de algoritmos que descomponen en factores un número. El algoritmo factoreo mas rápido que se conoce puede obtener un factor n en aproximadamente

$$\exp\sqrt{\ln(n)\cdot\ln(\ln(n))} = n^{\sqrt{\frac{\ln\ln(n)}{\ln(n)}}} = \ln(n)^{\sqrt{\frac{\ln(n)}{\ln(\ln(n))}}}$$

pasos (aca ln denota logaritmo natural). En la tabla 4.1 muestra la cantidad de operaciones que se necesitan para factorear n con el algoritmo más rápido, para

varias longitudes del número n (en dígitos decimales) y suponiendo que el tiempo requerido para cada operación es de un microsegundo.

Cantidad de Dígitos	Número de Operaciones	Tiempo de Procesamiento
50	1.4×10^{10}	3.9 horas
75	9.0×10^{12}	104 días
100	2.3×10^{15}	74 años
200	1.2×10^{23}	3.8×10^{9} años
300	1.5×10^{29}	4.9×10^{15} años
500	1.3×10^{39}	4.2×10^{25} años

Tabla 4.1 – Tiempo para "romper" por factorización el método RSA

Los autores del método recomiendan utilizan un número n de alrededor de 200 dígitos de longitud. Longitudes más largas o más cortas pueden depender de la importancia relativa de la velocidad y de la seguridad del cifrado. Un número de 80 dígitos provee una seguridad moderada contra ataques con la tecnología actual.

Esta flexibilidad de elegir una longitud de clave (y así un nivel de seguridad) para un determinado uso particular no es una característica encontrada en muchos métodos de encriptación.

4.1.3.2 Seguridad del algoritmo RSA

Aparte de factorizar n, podríamos intentar calcular $\phi(n)$ directamente, o probar por la fuerza bruta tratando de encontrar la clave privada. Ambos ataques son más costosos computacionalmente que la propia factorización de n.

4.1.3.3 Vulnerabilidades de RSA

Aunque el algoritmo RSA es bastante seguro conceptualmente, existen algunos puntos débiles en la forma de utilizarlo que pueden ser aprovechados por un atacante:

- **Claves Demasiado Cortas:** Actualmente se considera segura una clave RSA con una longitud de n de al menos 768 bits, si bien se recomienda el uso de claves no inferiores a 1024 bits. Se debe escoger la longitud de la clave, como se menciono anteriormente, en función del tiempo en que se quiere que la información permanezca en secreto.

- **Ataques de Intermediario:** El ataque de intermediario (figura 4.2) puede darse con cualquier algoritmo asimétrico. Supongamos que A quiere establecer una comunicación con B, y que C quiere espiarla. Cuando A le solicite a B su clave pública K_B, C se interpone, obteniendo la clave de B y enviando a A una clave falsa k_C creada por él. Cuando A codifique el mensaje, C lo interceptará de nuevo, decodificándolo con su clave propia y empleando K_B para recodificarlo y enviarlo a B. Ni A ni B son conscientes de que sus mensajes están siendo interceptados.

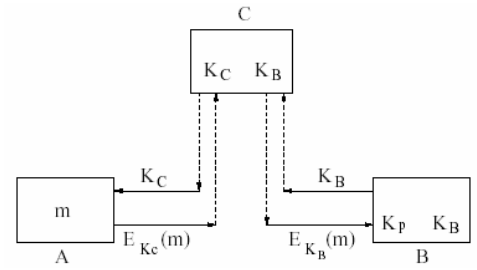

Figura 4.2 – Ataque de intermediario para un algoritmo asimétrico.

La única manera de evitar esto consiste en asegurar a A que la clave pública que tiene de B es auténtica. Para ello están los certificados de confianza, que certifican la autenticidad de la clave.

5 Firmas Digitales

Supongamos el caso en que dos personas tienen que validar su identidad pero no se conocen. Por ejemplo si una persona va a buscar un pasaje de avión a una agencia de viajes y tiene que acreditar su identidad, digamos que el pasaje esta a nombre de Juan Carlos Gómez, el empleado de la agencia de viajes acepta la identidad de quién dice ser Juan Carlos Gómez si cumple ciertos requisitos, por ejemplo si le muestra una identificación oficialmente válida (el pasaporte, el permiso de conducir, la cédula de identidad, etc.). Entonces el empleado compara la foto de la identificación con la apariencia del portador y decide si acepta el hecho que el cliente es quién dice ser.

La firma tradicional tiene varias características, la principal de ellas es que es aceptada legalmente, esto quiere decir que si alguna persona firmó un documento adquiere tanto los derechos como las obligaciones que de él deriven y si estas obligaciones no son acatadas, el portador del documento tiene el derecho de reclamación mediante un litigio. La autoridad competente acepta las responsabilidades adquiridas con sólo calificar a la firma como válida.

Se puede resumir en que existen dos procedimientos importantes, el primero el proceso de firma, que es el acto cuando una persona "firma" manualmente un documento. Y el proceso de verificación de la firma, que es el acto que determina si una firma es válida o no.

Por otro lado es importante hacer notar que la firma comprueba la identidad de una persona, de tal modo que así se sabe quién es la persona quién firmó, y ésta persona no puede negar las responsabilidades que adquiere en un documento firmado.

Resumiendo las ideas anteriores:

- *Proceso de firma:* este proceso es muy simple y consiste sólo en tomar un bolígrafo y estampar, dibujar o escribir garabatos en un papel. En general este garabato debe ser el mismo y es elegido a gusto de la persona. Se usa como una marca personal. Es importante mencionar que por un lado lo que identifica a la persona quien firma (quien hace el garabato) es la forma misma de la firma, pero también características de escritura, como la velocidad de escritura, la presión que se aplica al bolígrafo, la inclinación de la escritura, etc.

- *Proceso* de verificación: existen en general dos métodos de verificación de la firma, uno es el más usado y simple, que es el visual, esté método lo aplica cualquier cajero al pagar un cheque, o al efectuar un pago con tarjeta de crédito. En muchos casos la firma es rechazada por no pasar este método, sin embargo legalmente no es suficiente dicho método. El método legalmente definitivo es el peritaje de la firma en laboratorio, que consiste en verificar a la firma independientemente de la forma, tomando en cuenta otras características como la presión de escritura, la velocidad de escritura, la inclinación de escritura, las características particulares de alguna letra, etc. El conjunto de estas propiedades son propias de cada país y sus leyes. Es importante remarcar que el resultado es tomado como definitivo legalmente.

- Entonces con la firma queda resuelto legalmente el problema de la autenticidad o el comprobar la identidad de una persona.

- De la misma manera se soluciona el no-repudio, es decir, que una persona rechaza ser el autor de una firma.

- Es importante hacer notar que la firma frecuentemente se encuentra asentada en un documento de identidad oficialmente válido, como el pasaporte, el permiso de conducir, y otros.

El concepto de firma digital fue introducido por Diffie y Hellman en 1976. Básicamente una firma digital es un conjunto de datos asociados a un mensaje que permite asegurar la identidad del firmante y la integridad del mensaje. Aparentemente estos se consiguen con los criterios de autenticidad e integridad anteriormente mencionados, de manera tal de sustituir a la firma manuscrita. A continuación se mostrarán y analizarán distintos esquemas de firma digital.

5.1 Firma de clave privada

Un enfoque de las firmas digitales sería tener una autoridad central o arbitro que sepa todo y en quién todos confíen, al cual llamamos Big Brother (BB). Cada usuario escoge su clave privada y la lleva personalmente a las oficinas del BB. Por lo tanto, sólo Juan y el BB conocen la clave privada de Juan, es decir, K_j. Lo mismo realiza Carlos por su lado.

En la figura 5.1 se muestra el caso en que Juan le envía un mensaje a Carlos por un canal inseguro.

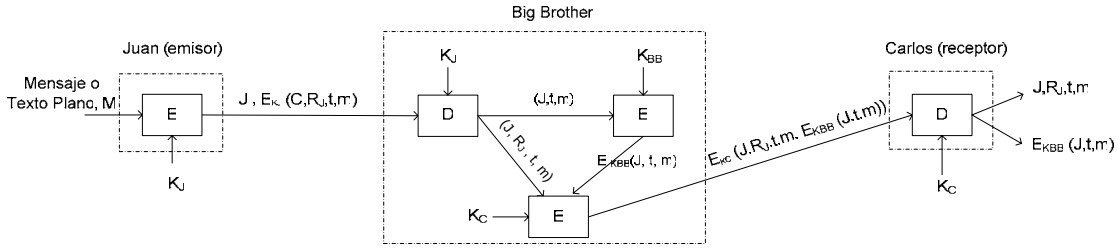

Figura 5.1 – Esquema Firma digital simétrica

Cuando Juan quiere enviarle un mensaje plano *m* a Carlos, genera $E_{KJ}(C,R_J,t,m)$ donde *C* es la identidad de Carlos, R_J es un número aleatorio elegido por Juan, *t* es una marca de tiempo para asegurar que el mensaje sea reciente, y $E_{KJ}(C,R_J,t,m)$ es el mensaje encriptado con su clave. A continuación lo envía como se muestra en la figura 5.1. El Big Brother ve que el mensaje es de Juan, lo desencripta, y envía un mensaje a Carlos como se muestra. El mensaje a Carlos contiene el mensaje plano de Juan y también el mensaje firmado $E_{KBB}(J,t,m)$. Ahora, Carlos tiene en su poder el mensaje enviado por Juan.

Supongamos que Juan niega posteriormente el mensaje enviado y para la solución de esta disputa se llega a juicio. El juez le pregunta a Carlos por qué está tan seguro de que el mensaje en disputa es proveniente de Juan y no de otra persona. Carlos primero indica que el Big Brother no aceptaría un mensaje de Juan

a menos que estuviera encriptado con la clave K_J, por lo que no hay posibilidad de que otra persona enviara al Big Brother un mensaje falso de Juan sin que él lo detectara de inmediato.

Carlos entonces presenta la prueba A, $E_{KBB}(J,t,m)$. Carlos dice que éste es un mensaje firmado por el Big Brother que comprueba que Juan envió m a él. El juez entonces pide que el Big Brother (en quien todo el mundo confía) que descifre la prueba A. Cuando él testifica que Carlos dice la verdad, el juez se pronuncia a favor de Carlos y caso cerrado.

Un problema potencial del protocolo de firma digital anterior es que una cuarta persona repita cualquiera de los dos mensajes. Para minimizar este problema, en todos los intercambios se usan marcas de tiempo. Es más, Carlos puede revisar todos los mensajes recientes para ver si se usó R_J en cualquiera de ellos. De ser así, el mensaje se descarta como repetición. De esta manera, Carlos rechazará los mensajes viejos con base en la marca de tiempo. Para protegerse contra ataques de repetición instantánea, Carlos simplemente examina el R_J de cada mensaje de entrada para ver si un mensaje igual se recibió de Juan durante la hora pasada. Si no, Carlos puede suponer con seguridad de que éste se trata de un nuevo mensaje.

5.2 Firma de clave publica

Un problema estructural del uso de la criptografía de clave simétrica para las firmas digitales es que todos tienen que confiar en el Big Brother. Es más, el Big Brother lee todos los mensajes firmados. Los candidatos más lógicos para operar el servidor del Big Brother son el gobierno, los bancos, los contadores y los abogados. Por desgracia, ninguna de estas organizaciones inspira confianza completa a todos los ciudadanos. Por lo tanto, sería bueno si la firma de documentos no requiere una autoridad confiable.

Afortunadamente, la criptografía de clave pública puede hacer una contribución importante aquí. Supongamos que los algoritmos públicos de encriptación y desencriptación tienen la propiedad de que E(D(m)) = m, además de la propiedad normal de D(E(m)) = m. Suponiendo que éste es el caso, Juan puede enviar un mensaje de texto plano firmado *m*, a Carlos transmitiendo $K_{puC}(K_{prJ}(m))$. Dado que solamente Juan conoce su clave privada $K_{prJ,}$ así como la clave pública de Carlos K_{puC}, por lo que Juan puede elaborar este mensaje.

Cuando Carlos recibe el mensaje, lo transforma usando su clave privada, como es normal, produciendo $K_{prJ}(m)$, como se muestra en la figura 5.2. Carlos almacena este texto en un lugar seguro y lo descifra usando la clave pública de Juan, obteniendo el mensaje plano original.

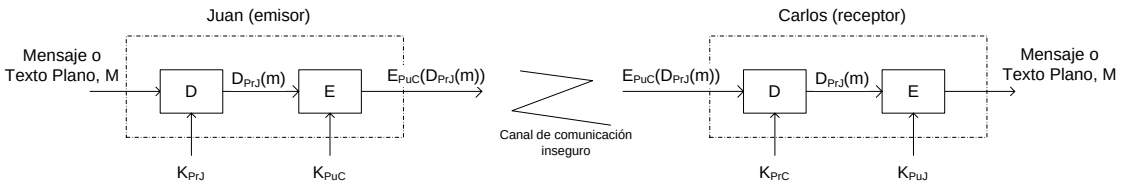

Figura 5.2 – Esquema Firma digital asimétrica

Para ver como funciona la propiedad de firma, supongamos que Juan posteriormente niega haber enviado el mensaje *m* a Carlos. Cuando el caso llega a juicio, Carlos puede presentar tanto *m* como $K_{prJ}(m)$. El juez puede comprobar fácilmente que Carlos tiene un mensaje válido encriptado con la clave privada de Juan con solo aplicar el proceso inverso al mismo. Puesto que Carlos no conoce la clave privada de Juan, la única forma en que Carlos puedo haber adquirido un mensaje encriptado con esa clave sería que Juan en efecto lo hubiera enviado. Mientras está en la cárcel por perjuicio y fraude, Juan tendrá tiempo suficiente para diseñar algoritmos de clave pública nuevos e interesantes.

Aunque el uso de criptografía de clave pública para las firmas digitales es un esquema elegante, hay problemas relacionados con el entorno en el que opera más que con el algoritmo básico. Por una parte, Carlos puede demostrar que un mensaje fue enviado por Juan siempre y cuando la clave privada de Juan permanezca en secreto. Si Juan la divulgara, el argumento ya no es válido, puesto que cualquiera pudo haber enviado el mensaje, inclusive el mismo Carlos.

Otro problema con el esquema de firmas es que ocurre si Juan decide cambiar su clave. Hacerlo ciertamente es legal, y probablemente es una buena idea cambiar la clave periódicamente. Si luego surge un caos en la corte, como se describió antes, el juez desencriptará $K_{prJ}(m)$ con la clave pública actual y no con la que firmo Juan originalmente. Entonces no se obtendrá el mensaje original *m*.

5.3 Función resumen

Una crítica a los métodos de firma es que con frecuencia combinan dos funciones dispares: autentificación y confidencialidad. En muchos casos se requiere la autentificación, pero no confidencialidad. Asimismo, con frecuencia la obtención de una licencia de exportación se facilita si el sistema en cuestión sólo proporciona autentificación pero no confidencialidad. A continuación se describirá un esquema de autentificación que no requiere la encriptación del mensaje completo.

Este esquema se basa en la idea de una función de resumen unidireccional (generalmente funciones de hash, como MD5 o SHA-1) que toma una parte arbitraria grande del texto plano y a partir de ella calcula una cadena de bits de longitud fija. Esta función de resumen r, tiene las siguientes cuatro propiedades importantes:

- Dado m, es fácil calcular r(m)
- Dado r(m), es imposible calcular m
- Dado m nadie puede encontrar m' de manera tal que r(m') = r(m)
- Un cambio a la entrada de incluso 1 bit produce una salida muy diferente

El cálculo de un resumen de un mensaje a partir de un trozo de texto plano es mucho más rápido que la encriptación de ese mensaje con un algoritmo de clave pública, por lo que los resúmenes de mensaje pueden usarse para acelerar los algoritmos de firma digital. Para ver su funcionamiento, revisando el protocolo de la figura 5.1, en lugar de firmar m con $E_{KBB}(J,t,m)$, el Big Brother ahora calcula el resumen del mensaje aplicándole la función r a m para producir r(m). Entonces

éste incluye $E_{KBB}(J,t,r(m))$ como elemento de lo encriptado que se envía a Carlos, en lugar de $E_{KBB}(J,t,m)$.

Si surge una disputa, Carlos puede presentar tanto m como $E_{KBB}(J,t,r(m))$. Una vez que el Big Brother lo desencripta para el juez, Carlos tiene r(m), que está garantizado que es genuino, y el m supuesto. Dado que es prácticamente imposible que Carlos encuentre otro mensaje que dé este resultado, el juez se convencerá fácilmente de que Carlos dice la verdad. Este uso de resúmenes de mensajes ahorra tanto tiempo de encriptación como costos de transporte de mensajes.

Los resúmenes de mensaje funcionan también en los criptosistemas asimétricos, como se muestra en la figura 5.3. Aquí, Juan primero calcula el resumen de su mensaje, luego lo firma (encripta con su clave privada) y lo envía a Carlos tanto el mensaje plano como el resumen firmado. En el caso de que una tercera persona reemplazará el mensaje m en el camino, Carlos se dará cuenta cuando calcule r(m) en el proceso de verificación.

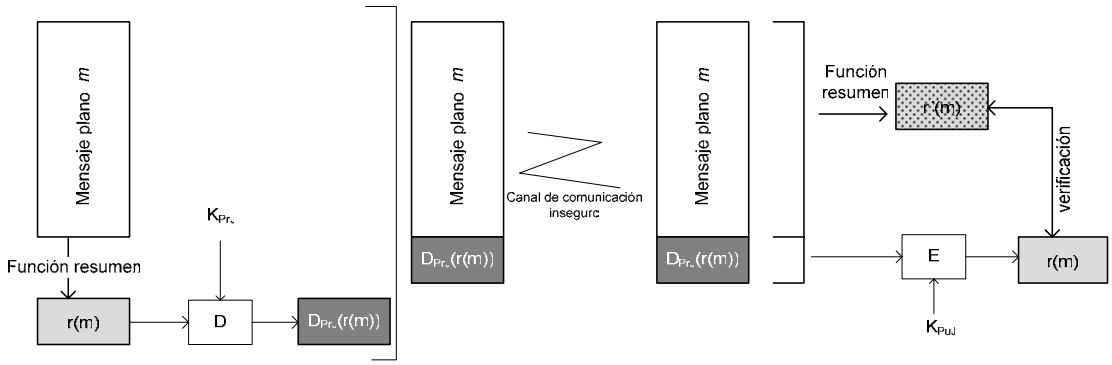

Figura 5.3 –Firma digital asimétrica utilizando resumen de mensaje

A continuación se mencionan algunas ventajas y desventajas de la firma digital con respecto a la manuscrita.

5.4 Ventajas

El procedimiento de verificación es exacto y es imposible en la práctica su falsificación.

Otra ventaja de la firma digital en su portabilidad, es decir, la firma digital puede ser realizada en diferentes partes del mundo, de forma simultánea sin la necesidad de testigos.

Una tercera ventaja que ofrece la firma digital es que esta puede ser compatible con los dispositivos electrónicos actuales. Es decir, el proceso de firma y de verificación son programas que pueden estar almacenados en una NoteBook, en una PC, etc.

5.5 Desventajas

Quizás la más notable desventaja actual de la firma digital contra la firma manuscrita, es que la primera no es válida legalmente aun en muchos países. Parece ser que esto obedece a una transición natural de esta nueva tecnología, que por lo tanto existe un rechazo en su aceptación a pesar de los grandes beneficios que proporciona.

Otra desventaja visible de la firma digital asimétrica es que su seguridad depende de la clave privada. Esto significa que si la clave privada se compromete por alguna causa, entonces se compromete la seguridad de la firma digital, ya que puede ser usada por individuos no autorizados.

Una desventaja más es que la firma digital esta cambiando conforma la tecnología avanza. Esto hace que ciertos documentos puedan ser comprometidos. Por lo que hay que crear métodos que permitan evitar esto.

6 Administración de claves públicas

La criptografía de clave pública hace posible que las personas que no comparten una clave común se comuniquen con seguridad. También posibilita firmar mensajes sin la presencia de una tercera persona de confianza. Por último, los resúmenes de mensajes firmados facilitan la verificación de la integridad de los mensajes.

Sin embargo, hay un problema que se ha pasado por alto: Si Juan y Carlos no se conocen entre sí, ¿cómo obtiene cada uno la clave pública del otro para iniciar el proceso de comunicación?. La solución obvia sería colocar su clave pública en su sitio Web, pero esta idea no funciona por la siguiente razón. Supongamos que Juan quiere buscar la clave pública de Carlos en el sitio web de él. ¿Cómo lo hace? Comienza tecleando la URL de Carlos. A continuación su navegador busca la dirección DNS de la página de inicio de Carlos y le envía una solicitud GET, como se muestra en la figura 6.1. Desgraciadamente, el intruso intercepta la solicitud y responde con una página de inicio falsa, probablemente una copia de la de Carlos, excepto por el reemplazo de la clave pública de Carlos con la del intruso. Cuando Juan encripta su primer mensaje con la clave K_{PUI}, el intruso lo desencripta, lo lee, lo vuelve a encriptar con la clave pública de Carlos y lo envía a éste, quien no tiene la menor idea de que hay un intruso que esta leyendo los mensajes que le llegan. Pero aún, el intruso puede modificar los mensajes antes de volverlos a encriptar para Carlos. Claramente, se necesita un mecanismo para asegurar que las claves públicas puedan intercambiarse de manera segura.

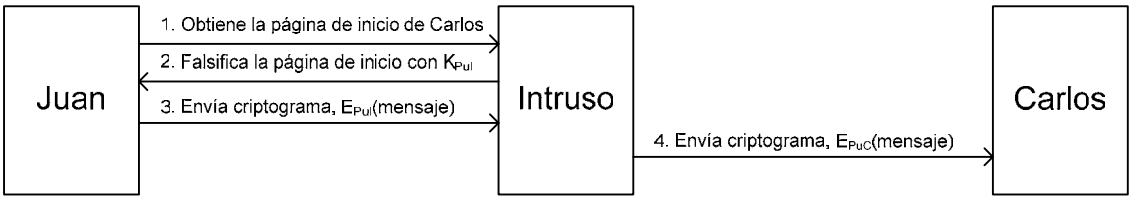

Figura 6.1 – Posible esquema donde un intruso sustituye la clave pública

6.1 Certificados

Como un primer intento para distribuir claves públicas de manera segura, se puede pensar en un centro de distribución de claves disponibles en línea las 24 horas del día que proporciona claves pública a petición. Uno de los muchos problemas con esta solución es que no es escalable, y el centro de distribución de claves podría volver rápidamente en un cuello de botella. Además, si alguna vez fallara, la seguridad en Internet podría reducirse a nada.

Por estas razones, se ha desarrollado una solución diferente, una que no requiere que el centro de distribución esté en línea todo el tiempo. De hecho, ni siquiera tiene que estar en línea. En su lugar, lo que hace es certificar las claves públicas que pertenecen a las personas, empresas y otras organizaciones. Una organización que certifica claves públicas se conoce como *AC* (Autoridad de Certificación).

Como un ejemplo, suponga que Carlos desea permitir que Juan y otras personas se comuniquen con él de manera segura. Él puede ir a la *AC* con su clave pública junto con su pasaporte o permiso de conducir para pedir su certificación. A continuación, la *AC* emite un certificado similar al que se muestra en la figura 6.2 y firma el resumen del certificado con su clave privada (como función de resumen se podría utilizar la función de hash SHA-1). Carlos paga la cuota de la *AC* y obtiene el certificado y su resumen firmado.

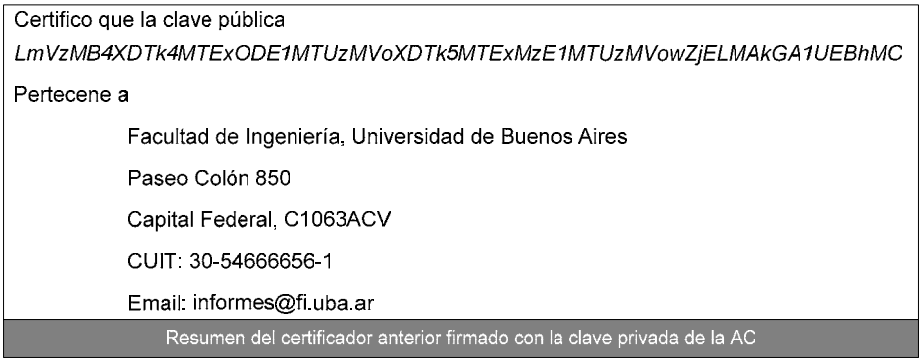

Figura 6.2 – Ejemplo de certificado digital y su firma

El trabajo fundamental de un certificado es relacionar una clave pública con el nombre de un personaje principal (individual, empresa, etcétera). Los certificados mismos no son secretos ni protegidos. Por ejemplo, Carlos podría decidir colocar su nuevo certificado en su sitio Web, con un vínculo en la página de inicio que diga: Haga click para obtener mi certificado de clave pública. El click resultante podría regresar el certificado y el resumen de la firma.

Volviendo al escenario que se muestra en la figura 6.1. Cuando el intruso intercepta la solicitud que Juan realiza para obtener la página de inicio de Carlos, ¿qué puede hacer el intruso?. Puede poner su propio certificado y el resumen firmado falsificado, pero cuando Juan lea el certificado, verá inmediatamente que no esta hablando con Carlos porque el nombre de éste no se encuentra en dicho certificado. El intruso puede modificar sobre la marcha la página de inicio Carlos, reemplazando la clave pública de Carlos con la suya. Sin embargo, cuando Juan ejecute la función resumen en el certificado, obtendrá un resumen que no corresponde con el que obtuvo cuando aplico la clave pública de la *AC* al resumen firmado por él. Puesto que el intruso no tiene la clave privada de la *AC*, no tiene forma de generar un resumen firmado que contenga el resumen de a página Web modificada con su clave pública en él. De esta manera, Juan puede estar seguro de que tiene la clave pública de Carlos y no la de otra persona. Además este esquema no requiere que la *AC* esté en línea para la verificación, por lo tanto se elimina un cuello de botella potencial.

Mientras que la función estándar de un certificado es relacionar una clave pública a un personaje principal, un certificado también se puede utilizar para relacionar una clave pública a un *atributo*. Por ejemplo, un certificado podría decir: Esta clave pública pertenece a alguien mayor de 18 años. Podría utilizarse para probar que el dueño de la clave privada no es una persona menor de edad y por lo tanto, se le permitió acceder a material no apto para niños, entre otras cosas, pero sin revelar la identidad del dueño. Por lo general, la persona que tiene el certificado podría enviarlo al sitio Web, al personaje principal o al proceso que se preocupa por

la edad. El sitio, el personaje principal o el proceso podrían generar a continuación un número aleatorio y encriptado con la clave pública del certificado. Si el dueño pudiera desencriptarlo y regresarlo, esa sería una prueba de que el dueño tenía el atributo establecido en el certificado. De manera alternativa, el número aleatorio podría utilizarse para generar una clave de sesión para la conversación resultante.

Apéndice A – Cómo elegir contraseñas

La contraseña es tal vez la parte más vulnerable de un sistema informático. Aun el sistema más seguro puede caer en manos de un atacante si éste logra ingresar a causa de una contraseña mal elegida.

Con el uso cada vez más difundido de operaciones electrónicas donde el usuario debe primero demostrar que es quien dice ser y que está autorizado para efectuarlas (autenticación), una mala contraseña puede llevar no solamente a robo o pérdidas sino a responsabilidad legal frente a terceros.

A continuación se listan algunas pautas para la buena elección de contraseñas:

- NO debes utilizar tu nombre de usuario o algún derivado (invertido, en mayúsculas/minúsculas o con otros caracteres.) Un error demasiado común es elegir como password el nombre de usuario con unos números, por ejemplo el año: carlos99, alfa00, etc.
- NO debes usar tu nombre y/o apellido o algún derivado de éstos.
- NO debes emplear el nombre de tu esposa o de sus hijos (en caso de tener).
- NO debes utilizar tus datos personales que se puedan obtener fácilmente, como por ejemplo, número de placa o marca del vehículo, números telefónicos, números de identificación, la dirección de su casa, etc.
- NO debes emplear una contraseña que contenga solamente números o la misma letra repetida.
- NO debes emplear una palabra del idioma Español o de otro idioma.
- NO debes usar una contraseña de menos de seis caracteres.
- SI debes emplear una mezcla de mayúsculas y minúsculas.
- SI debes emplear una contraseña que contenga caracteres no alfabéticos (números o signos de puntuación, para estos últimos, es conveniente averiguar cuáles permite su sistema particular.)
- SI debes escoger una contraseña fácil de recordar para no tener que escribirla.
- SI debe escoger una contraseña que pueda escribir rápidamente, sin mirar el teclado.

Referencias

Biham, E. Anderson, R. Knudsen, L. Serpent. 1998. *A New Block Cipher Proposal*.

Burwick, C. Coppersmith, D. D´Avignon, E. Gennaro, R. Halevi, S. Jutla, C. Matyas, S. O´Connor, L. Peyravian, M. Safford, D. Zunic, N. 1999. *MARS - a candidate cipher for AES*. IBM Corporation.

Daemen, J. Rijmen, V. 1998. *AES Proposal: Rijndael*.

Lucena López, M. 2002. *Criptografía y Seguridad en Computadores*. Tercera Edicion. Versión 1.14

Rivest, R. Robshaw, M. Sidney, R. Yin, Y. 1998. *The RC6 Block Cipher*. MIT Laboratoy for Computer Science. 545 Tecnology Square. Cambridge. MA 02139. USA.

Rivest, R. Shamir, A. Adleman, L. 1978. *A Method for Obtaining Digital Signatures and Public-Key Crytosystems*.

Robshaw, M. 1995a. *Block Ciphers*. RSA Laboratories Technical Report TR-601. Version 2.0.

Robshaw, M. 1995b. *Stream Ciphers*. RSA Laboratories Technical Report TR-701. Version 2.0.

Stinson, D. 1995. *Cryptography: Theory and Practice.* CRC Press.

Schneier, B. 1996. *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C.* Second Edition. Wiley Computer Publishing. John Wiley & Sons, Inc.

Shneier, B. Kelsey, J. Whiting, D. Wagner, D. Hall, C. Ferguson, N. 1998. *Twofish: A 128-Bit Block Cipher*.

Tanenbaum, A. 2003. *Computer Networks.* Fourth Edition. Prentice Hall.