Tema 8: Introducción a la criptografía.

8.1 Introducción.

En su concepción inicial y en sus primeros usos, las redes de ordenadores fueron usadas generalmente para el envío de correo electrónico y para compartir recursos, generalmente impresoras, en empresas de mediano/gran tamaño.

En estas condiciones la seguridad de la información que circulaba por esas redes carecía prácticamente de importancia y no fue objeto de atención. Sin embargo, en la actualidad millones de personas usan las redes informáticas para transacciones bancarias, compras, etc., con lo que la seguridad aparece como una necesidad a cubrir.

Los problemas de seguridad de las redes pueden dividirse de forma general en cuatro áreas interrelacionadas:

- El secreto, encargado de mantener la información fuera de las manos de usuarios no autorizados.

- La validación de identificación, encargada de determinar la identidad de la persona u ordenador con el que se establece una comunicación.

- El control de integridad, encargado de asegurar que un mensaje recibido es recibido con el contenido enviado por la otra parte, y no un mensaje manipulado por un tercero.

- El no repudio, encargado de asegurar la "firma" de los mensajes, de igual forma que se firma en papel cualquier operación realizada por las personas, como pueden ser una operación de compra/venta, la firma de las notas de un examen, etc.

Aunque muchos de estos problemas tratan de resolverse en capas de la red que se encuentran por debajo de la capa de aplicación, por ejemplo en la capa de red pueden instalarse muros de seguridad para mantener adentro (o afuera) los paquetes, en la capa de transporte pueden cifrarse conexiones enteras terminal a terminal, ninguna de ellas resuelve completamente los problemas de seguridad antes enumerados.

La resolución de estos problemas de seguridad se realiza como una parte previa o de apoyo de la capa de aplicación. A continuación se exponen distintas soluciones a los problemas planteados con anterioridad, esto es, el secreto, la validación de identificación, el control de integridad y el no repudio.

8.2 Resolución del problema de seguridad del secreto.

La resolución del problema del secreto en la red (y del secreto de los mensajes en cualquier sistema de comunicación), ha estado siempre unido al cifrado (codificación) de los mensajes.

Hasta la llegada de las computadoras, la principal restricción del cifrado consistía en la capacidad del empleado encargado de la codificación para realizar las transformaciones necesarias y en la dificultad de cambiar rápidamente el método de cifrado, pues esto implicaba entrenar a una gran cantidad de personas[1].

Los mensajes a cifrar, conocidos como texto normal, se transforman mediante una función parametrizada por una clave. La salida del cifrado, conocida como texto cifrado, es transmitida después. Si un intruso escucha y copia el texto cifrado, a diferencia del destinatario original, no conoce la clave de cifrado y no puede descifrar fácilmente el texto cifrado. El arte de descifrar se llama criptoanálisis y la persona que descifra mensajes cifrados se conoce como criptoanalista. El arte de diseñar cifradores se conoce como criptografía y a la unión de ambos se la conoce como criptología.

A partir de aquí usaremos $C=E_k(P)$ para indicar que el cifrado del texto normal P usando la clave K da el texto cifrado C. Del mismo modo $P=D_k(C)$ representa el descifrado de C para obtener el texto normal nuevamente, por lo que $D_k(E_k(P))=P$. Esta notación sugiere que E y D son sólo funciones matemáticas de dos parámetros, de los cuales hemos escrito uno (la clave) como subíndice, en lugar de como argumento, para distinguirlo del mensaje.

Actualmente, las reglas fundamentales de la criptografía consiste en suponer que el criptoanalista conoce el método general de cifrado usado, esto es, el criptoanalista conoce E, pues la cantidad de esfuerzo necesario para inventar, probar e instalar un método nuevo cada vez que el viejo es conocido hace impracticable mantenerlo en secreto, y que no conoce la clave, que consiste en una cadena relativamente corta que selecciona uno de los muchos cifrados potenciales y que puede ser cambiada de forma sencilla con la frecuencia deseada[2].

8.2.1 Rellenos de una sola vez.

La construcción de un cifrado inviolable es bastante sencilla. La técnica se conoce desde hace décadas y consiste en escoger una cadena de bits al azar como clave. Luego se convierte el texto normal en una cadena de bits, por ejemplo usando su representación ASCII. Por último, se calcula el or exclusivo (XOR) y cuya tabla de valores lógicos puede verse en la siguiente figura, de estas dos cadenas, bit por bit.

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Figura 8.2.1.1: Tabla lógica de la función o-exclusivo (XOR).

[1] La necesidad de cambiar el método de cifrado se hace evidente si se tienen en cuenta hechos como la capacidad de los Aliados, durante la Segunda Guerra Mundial, de descifrar el código de cifrado Enigma utilizado por el ejército alemán durante esa contienda.

[2] Un ejemplo sencillo es una cerradura de combinación. Todo el mundo conoce como funciona, pero la clave es secreta. Una longitud de clave de tres dígitos significa que existen 1000 posibilidades, una longitud de clave de seis dígitos implica un millón de posibilidades.

El texto cifrado resultante no puede descifrarse porque cada texto normal posible es un candidato igualmente probable. El texto cifrado no proporciona al criptoanalista ninguna información en absoluto. En una muestra suficientemente grande de texto cifrado, cada letra ocurrirá con la misma frecuencia, al igual que cada digrama (combinación de dos letras) y cada trigrama (combinación de tres letras). Como ejemplo, cifremos el mensaje "texto cifrado" con la cadena "En un lugar de la Mancha de cuyo nombre…"

Texto original	t	e	x	t	o		c	i	f	r	a	d	o
Codificación ASCII (hex)	74	65	78	74	6F	20	63	69	66	72	61	64	6F
Texto de cifrado	E	n		u	n		l	u	g	a	r		d
Codificación ASCII (hex)	45	6E	20	75	6E	20	6C	75	67	61	72	20	64
Codificación cifrada (hex)	*31*	*0B*	*58*	*01*	*01*	*00*	*0F*	*1C*	*01*	*13*	*13*	*44*	*08*

Figura 8.2.1.2: Cifrado de un texto mediante relleno de una sola vez.

Si procedemos ahora a descifrarlo con la clave de codificación, obtenemos el mensaje original:

Codificación cifrada (hex)	*31*	*0B*	*58*	*01*	*01*	*00*	*0F*	*1C*	*01*	*13*	*13*	*44*	*08*
Texto de cifrado	E	n		u	n		l	u	g	a	r		d
Codificación ASCII (hex)	45	6E	20	75	6E	20	6C	75	67	61	72	20	64
Codificación ASCII (hex)	74	65	78	74	6F	20	63	69	66	72	61	64	6F
Texto original	t	e	x	t	o		c	i	f	r	a	d	o

Figura 8.2.1.3: Descifrado de un texto cifrado mediante relleno de una sola vez.

Sin embargo, este método tiene varias desventajas prácticas. En primer lugar, la clave no puede memorizarse, por lo que tanto el transmisor como el receptor deben llevar una copia por escrito consigo. Además, la cantidad total de datos que pueden transmitirse está limitada a la cantidad de clave disponible. Otro problema es la sensibilidad del método a la pérdida o inserción de caracteres. Si el transmisor y el receptor pierden la sincronía, todos los datos a partir de ahí aparecerán alterados.

8.3 Criptografía clásica.

La criptografía clásica se basa en algoritmos sencillos y claves muy largas para la seguridad. Las técnicas criptográficas clásicas son básicamente dos, el cifrado por sustitución y el cifrado por trasposición[3].

8.3.1 Cifrado por sustitución.

El cifrado por sustitución se basa en la sustitución de cada letra o grupo de letras por otra letra o grupo de letras para disfrazarla. Uno de los cifrados por sustitución más antiguos conocidos es el cifrado de Cesar, atribuido al emperador romano Julio Cesar. En este método la letra *a* se convierte en *D*, la *b* en *E*, la *c* en *F*, ... , y *z* se vuelve *C*. Así, el mensaje *ataque* se convierte en *DWDTXH*. Una generalización del cifrado de Cesar permite que el alfabeto de texto cifrado se desplaza k letras en lugar de siempre 3, con lo cual k se convierte en la clave de cifrado.

[3] A partir de este momento, tomaremos como texto normal aquel que se encuentra representado por letras minúsculas y como texto cifrado el que se encuentre representado por letras mayúsculas.

La siguiente mejora es hacer que cada uno de los símbolos del texto normal, por ejemplo las 26 letras del alfabeto castellano si omitimos la letra ñ, tengan una correspondencia biunívoca con alguna otra letra[4]. Por ejemplo:

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

Este sistema general se llama sustitución monoalfabética, siendo la clave la cadena de 26 letras correspondiente al alfabeto completo. A primera vista, esto podría parecer un sistema seguro, porque aunque el criptoanalista conoce el sistema general (sustitución letra por letra), no sabe cuál de las 26! = $4x10^{26}$ claves posibles se está usando. Sin embargo, si se cuenta con una cantidad pequeña de texto cifrado, el cifrado puede descifrarse fácilmente. El ataque básico aprovecha las propiedades estadísticas de los lenguajes naturales.

En castellano, la letra *e* es la más común, seguida de *a*, *o*, *l*, *s*, *n*, *d*, etc. Las combinaciones más comunes de dos letras o digramas son *de*, *la*, *el*, *en*, *se*, *un*, *no*, *su*, *al*, *es*, etc. Las combinaciones más comunes de tres letras o trigramas son *que*, *los*, *del*, *las*, *por*, *con*, *una*, *mas*, *sus*, *han*, etc.

Un criptoanalista que intenta descifrar una codificación monoalfabética comenzaría por contar la frecuencia relativa de todas las letras del texto cifrado. Entonces podría asignar tentativamente la más común a la letra *e* y la siguiente más común a la letra *a*. Vería entonces los digramas y trigramas que aparecen para encontrar correspondencias con los más habituales y de esta forma descifraría el texto.

Como ejemplo, veamos el siguiente, escrito en castellano, y cifrado con una codificación monoalfabética:

```
Q DTRORQ JXT SGL QHXFZTL RT ZTGKOQ, TFXFEOQRGL RT HKQEZOEQL,
TZE., LTQF TSQWGKQRGL, LT EGSGEQKQF TF TS EGKKTLHGFROTFZT
QHQKZQRG. HQKQ CTK SGL XSZODGL RGEXDTFZGL HXTLZGL TF SQ HQUOFQ
COLOZQ TS QHQKZQRG RT QCOLGL.
```

Para descifrar este cifrado monoalfabético, tomemos las tablas que nos indican la frecuencia de aparición de las letras, los digramas y los trigramas en castellano:

Letra	Frecuencia	Letra	Frecuencia	Letra	Frecuencia	Letra	Frecuencia
e	16.78	a	11.96	o	8.69	l	8.37
s	7.88	n	7.01	d	6.87	r	4.94
u	4.80	i	4.15	t	3.31	c	2.92
p	2.76	m	2.12				

Digrama	Frecuencia	Digrama	Frecuencia	Digrama	Frecuencia
de	778	la	460	el	339
en	302	se	119	un	98
no	74	su	64	al	63
es	47				

[4] Existen cifrados por sustitución que utilizan varias correspondencias biunívocas, pero basta con identificar el número de correspondencias realizadas para descifrar el texto como veremos a continuación.

Trigrama	Frecuencia	Trigrama	Frecuencia	Trigrama	Frecuencia
que	289	los	196	del	156
las	114	por	110	con	82
una	78	mas	36	sus	27
han	19				

Si analizamos la frecuencia de aparición de cada letra en nuestro texto cifrado obtenemos:

Letra	Frecuencia	Letra	Frecuencia	Letra	Frecuencia	Letra	Frecuencia
Q	11.85	T	10.43	G	8.06	L	7.58
F	5.21	R	5.21	Z	5.21	K	4.74
O	4.74	E	3.79	H	3.79	S	3.79
X	2.84	C	1.42	D	1.42	J	0.47
U	0.47	W	0.47				

Si analizamos la frecuencia de aparición de las letras en nuestro texto y la frecuencia normal en castellano, podemos deducir que la *Q* se corresponde con la *e* y la *T* con la *a* o viceversa.

Analizando los digramas, vemos que los digramas correspondientes a la letra *e*, como son *de*, *el*, *en*, *se*, etc., se corresponden con aquellos que contienen la letra *T*, con lo cual deducimos que la *T* corresponde con la *e* y la *Q* con la *a*. Realizando esta sustitución, obtenemos:

```
a DeRORa JXe SGL aHXFZeL Re ZeGKOa, eFXFEOaRGL Re HKaEZOEaL,
eZE., LeaF eSaWGKaRGL, Le EGSGEaKaF eF eS EGKKeLHGFROeFZe
aHaKZaRG. HaKa CeK SGL XSZODGL RGEXDeFZGL HXeLZGL eF Sa HaUOFa
COLOZa eS aHaKZaRG Re aCOLGL.
```

Si seguimos analizando los digramas, podemos ver que el digrama *Re* se repite con mucha frecuencia, por lo que podemos suponer razonablemente que la *R* es la *d* y realizar la sustitución:

```
a DedOda JXe SGL aHXFZeL de ZeGKOa, eFXFEOadGL de HKaEZOEaL,
eZE., LeaF eSaWGKadGL, Le EGSGEaKaF eF eS EGKKeLHGFdOeFZe
aHaKZadG. HaKa CeK SGL XSZODGL dGEXDeFZGL HXeLZGL eF Sa HaUOFa
COLOZa eS aHaKZadG de aCOLGL.
```

Mirando ahora la frecuencia de aparición de los trigramas en castellano, podemos ver que el de mayor frecuencia es que, y tenemos un trigrama *JXe* en nuestro texto. Suponiendo que *J* es *q* y que *X* es *u*, obtenemos:

```
a DedOda que SGL aHuFZeL de ZeGKOa, eFuFEOadGL de HKaEZOEaL,
eZE., LeaF eSaWGKadGL, Le EGSGEaKaF eF eS EGKKeLHGFdOeFZe
aHaKZadG. HaKa CeK SGL uSZODGL dGEuDeFZGL HueLZGL eF Sa HaUOFa
COLOZa eS aHaKZadG de aCOLGL.
```

Continuando el análisis de los trigramas, vemos que aparece varias veces el trigrama *SGL*, que debería corresponderse con el trigrama castellano *los*, y que además

no hemos obtenido todavía la equivalencia de esas letras. Comprobando que la frecuencia de aparición de dichas letras se corresponde a la esperada en castellano, y realizando la sustitución, obtenemos:

```
a DedOda que los aHuFZes de ZeoKOa, eFuFEOados de HKaEZOEas,
eZE., seaF elaWoKados, se EoloEaKaF eF el EoKKesHoFdOeFZe
aHaKZado. HaKa CeK los ulZODos doEuDeFZos HuesZos eF la HaUOFa
COsOZa el aHaKZado de aCOsos.
```

Continuando el análisis, podemos ver que el digrama *en* es bastante frecuente y no ha aparecido, teniendo el texto la aparición de *eF* y comprobando que la frecuencia de aparición de *F* se corresponde aproximadamente con la de la letra *n*, haciendo la sustitución *F* por *n* obtenemos:

```
a DedOda que los aHunZes de ZeoKOa, enunEOados de HKaEZOEas,
eZE., sean elaWoKados, se EoloEaKan en el EoKKesHondOenZe
aHaKZado. HaKa CeK los ulZODos doEuDenZos HuesZos en la HaUOna
COsOZa el aHaKZado de aCOsos.
```

Como el texto actual es más o menos legible, es sencillo tomar algunas decisiones como suponer que la *Z* es la *t*, la *O* es la *i*, la *H* es la *p*, la *E* es la *c* y la *K* es la *r*, comprobando que las frecuencias de aparición se corresponden con las esperadas. Realizando estas sustituciones:

```
a Dedida que los apuntes de teoria, enunciados de practicas,
etc., sean elaWorados, se colocaran en el correspondiente
apartado. para Cer los ultiDos docuDentos puestos en la paUina
Cisita el apartado de aCisos.
```

Los pocos elementos que faltan por obtener, son fácilmente identificables, por ejemplo, la *D* corresponde a la *m* para hacer la palabra *medida*, etc. Realizando estos cambios finales, obtenemos el siguiente texto descifrado:

```
a medida que los apuntes de teoria, enunciados de practicas,
etc., sean elaborados, se colocaran en el correspondiente
apartado. para ver los ultimos documentos puestos en la pagina
visita el apartado de avisos.
```

Otro enfoque posible para descifrar el cifrado por sustitución es adivinar una palabra o frase probable del mensaje. Por ejemplo, en un mensaje de una compañía contable es muy probable la aparición de la palabra *contabilidad*. Usando nuestro conocimiento de que *contabilidad* tiene la letra *a* repetida, con cinco letras intermedias, entre ellas, en las posiciones 2 y 4, la letra *i* repetida con una letra entre ella, y la letra *d* al final repetida con la letra *a* entre ellas, podemos buscar en el mensaje fragmentos que correspondan a esta estructura y obtener el texto sin cifrar.

8.3.2 Cifrados por trasposición.

Los cifrados por sustitución conservan el orden de los símbolos de texto normal, pero los disfrazan. Los cifrados por trasposición en contraste, reordenan las letras pero no las disfrazan.

Un ejemplo de cifrado por trasposición es la trasposición en columnas. La clave del cifrado es una palabra o frase que no contiene letras repetidas y que indica el orden en que han sido alteradas las palabras. Por ejemplo, supongamos el siguiente texto sin cifrar:

```
en junio tenemos examen de atd
```

Si utilizamos ahora como clave de cifrado la palabra *peso*, realizamos el siguiente proceso:

p	e	s	o
3	1	4	2
e	n	j	u
n	i	o	t
e	n	e	m
o	s	e	x
a	m	e	n
d	e	a	t
d	x	x	x

Donde las x del final se han puesto para completar la fila. Reordenando el texto en función de la clave, obtenemos el siguiente texto cifrado:

```
NINSMEXUTMXNTXENEOADDJOEEEAX
```

Para descifrar un cifrado por trasposición, el criptoanalista debe primero ser consciente de que está tratando con un cifrado por trasposición. Observando la frecuencia de aparición de las letras, puede comprobar que se ajustan al patrón usual del texto normal. De ser así, es evidente que se trata de un cifrado por trasposición, pues en tal cifrado cada letra se representa a sí misma.

El siguiente paso es intentar obtener la cantidad de columnas. En muchos casos, puede suponerse una palabra o frase probable, por el contexto del mensaje. Por ejemplo, en nuestro caso, si sospechamos que aparece la palabra junio, podemos buscar la cantidad de columnas que existen, pues si son dos debe aparecer JNO y UI, si son tres JI y UO y si son cuatro debe aparece JO. Comprobando el texto cifrado, vemos que aparece JO, con lo que el número de columnas es de cuatro. Distribuyendo el texto en cuatro columnas tenemos:

N	U	E	J
I	T	N	O
N	M	E	E
S	X	O	E
M	N	A	E
E	T	D	A
X	X	D	X

El paso restante es ordenar las columnas. Cuando la cantidad de columnas, k, es pequeña, puede examinarse cada uno de los pares de columnas k(k-1) para ver si la frecuencia de sus diagramas es igual a la del texto normal. El par con mejor concordancia se supone correctamente ubicado. Ahora cada columna restante se prueba tentativamente como el sucesor de este par. La columna cuyas frecuencias de digramas

y trigramas produce la mejor concordancia se toma tentativamente como correcta. La columna antecesora se encuentra de la misma manera. El proceso completo se repite hasta encontrar un orden potencial. Es probable que el texto normal sea reconocible en algún punto y permita realizar la decodificación del mismo.

En nuestro ejemplo, como sabemos que JO corresponde a junio y dado que nuestra clave es pequeña, podemos buscar las posibilidades de combinación de las columnas. En concreto, la palabra junio puede aparecer como:

```
J U N I       J U N       J U           J
O           I O       N I O       U N I O
```

Como en nuestro caso en la primera fila no existe ninguna letra I, sabemos que la primera posibilidad no es valida. Además, como en la primera fila tenemos una U y no existe tal letra en la segunda, sabemos que la última posibilidad tampoco es valida. Rescribiendo nuestro texto, tenemos:

```
N E J U
I N O T
N E E M
S O E X
M A E N
E D A T
X D X X
```

Ahora debemos tener en cuenta las dos posibilidades que nos quedan, probando ambas:

```
N E J U  →  E J U N    E N J U
I N O T  →  N O T I    N I O T
N E E M  →  E E M N    E N E M
S O E X  →  O E X S    O S E X
M A E N  →  A E N M    A M E N
E D A T  →  D A T E    D E A T
X D X X  →  X X D X    D X X X
```

Pudiendo comprobar que la solución correcta es la segunda.

8.4 Criptografía moderna.

La criptografía moderna usa las mismas ideas básicas que la criptografía tradicional, la trasposición y la sustitución, pero su orientación es distinta. Mientras la criptografía tradicional usaba algoritmos sencillos y claves muy largas para la seguridad, hoy en día es cierta la afirmación contraria: el objetivo es hacer algoritmos de cifrado tan complicados y rebuscados que incluso si el criptoanalista obtiene cantidades enormes de texto cifrado a su gusto, no será capaz de entender nada y por tanto de descifrarlo.

Las trasposiciones y sustituciones pueden implantarse mediante circuitos sencillos. En la figura siguiente se muestran dos dispositivos conocidos como caja P, que se usa para efectuar una trasposición de una entrada de 12 bits; y otro dispositivo

conocido como caja S, en el cual ingresa un texto normal de 3 bits y sale un texto cifrado de 3 bits. La potencia real de estos elementos básicos sólo se hace aparente cuando ponemos en cascada una serie completa de estas cajas para formar un cifrado de producto como podemos ver en la figura siguiente.

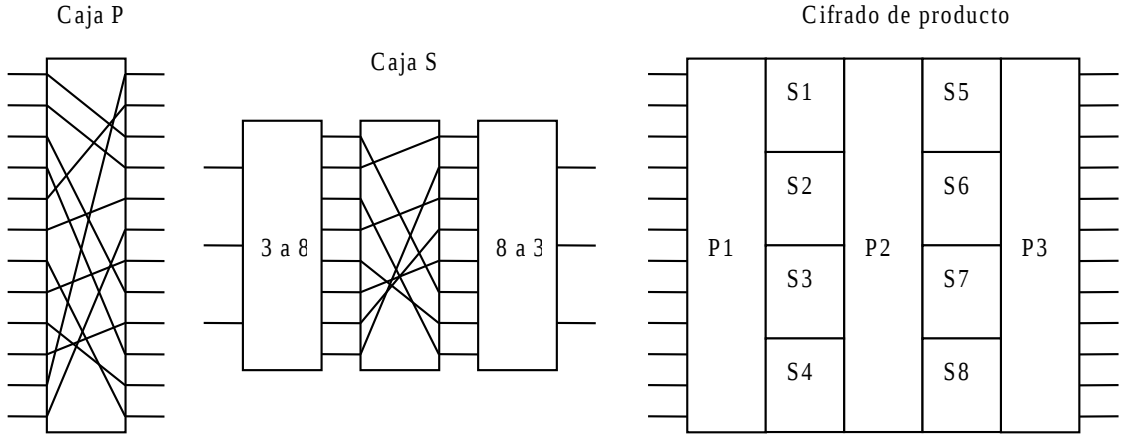


Figura 8.4.1: Ejemplo de cifrado de producto mediante cajas P y cajas S.

El cifrado moderno se divide actualmente en cifrado de clave privada y cifrado de clave pública. En el cifrado de clave privada las claves de cifrado y descifrado son la misma (o bien se deriva de forma directa una de la otra), debiendo mantenerse en secreto dicha clave para evitar el descifrado de una mensaje. Por el contrario, en el cifrado de clave pública las claves de cifrado y descifrado son independientes, no derivándose una de la otra, por lo cual puede hacerse pública la clave de cifrado siempre que se mantenga en secreto la clave de descifrado. Veremos a continuación algunos algoritmos de cifrado modernos.

8.4.1 Cifrado DES (Data Encryption Standar).

El algoritmo DES es un algoritmo de cifrado de clave privada desarrollado por IBM a principios de la década de los 70 a partir de otro algoritmo conocido como Lucifer que utilizaba claves de 112 bits y que fueron reducidas a 56 bits en el algoritmo DES. La reducción del tamaño de las claves, propuesta por la NSA (Agencia Nacional de Seguridad) originó controversia debido a que sé pensó que la NSA había debilitado intencionadamente el algoritmo del DES para poder desencriptarlo.

Dicha controversia llegó a su fin cuando en 1994 IBM publicó un artículo describiendo los criterios del desarrollo del algoritmo DES. El artículo indica que el diseño se realizó de forma que el algoritmo DES fuera resistente a criptoanálisis, pero lo suficientemente sencillo como para poder ser implementado en un circuito electrónico con la tecnología de principios de los años 70. Por tanto, el algoritmo DES se diseño de forma que no pudiera ser descifrado por criptoanálisis, pero si que puede ser descifrado probando todas las claves posibles, asumiendo que se cuenta con el hardware adecuado.

En la siguiente figura se muestra un esbozo del algoritmo DES. El texto normal se cifra en bloques de 64 bits, produciendo 64 bits de texto cifrado. El algoritmo, que se parametriza con la clave de 56 bits, tiene 19 etapas diferentes.

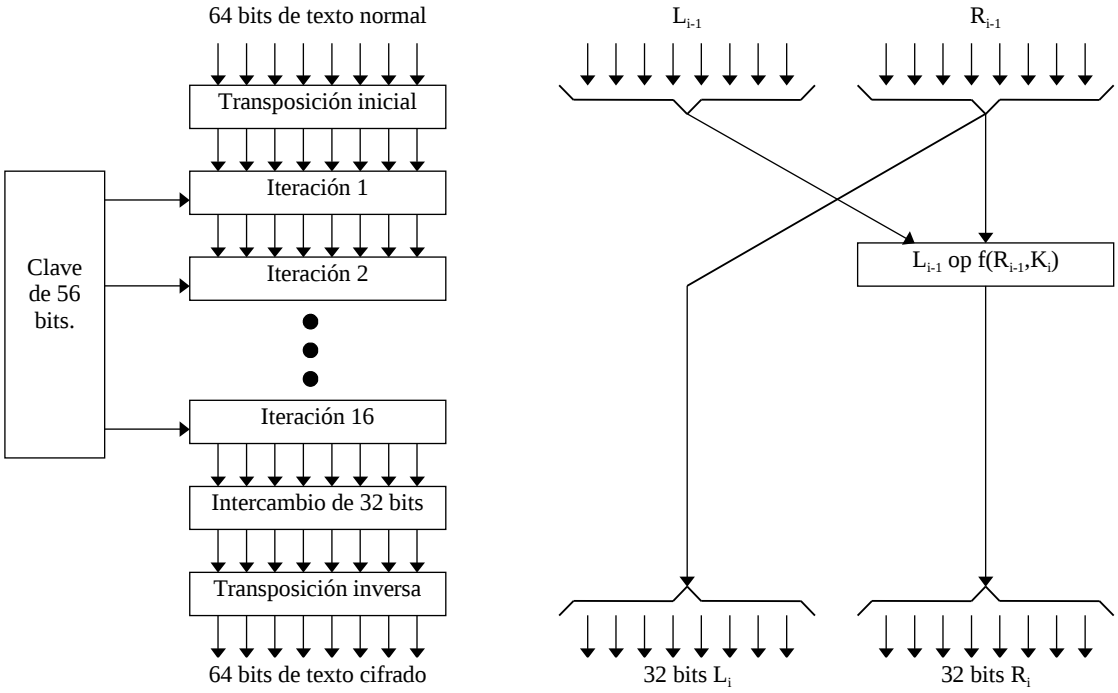


Figura 8.4.1.1: Algoritmo del cifrado DES.

La primera etapa es una trasposición, independiente de la clave, del texto normal de 64 bits. La última etapa es el inverso exacto de esta trasposición. La etapa previa a la última intercambia los 32 bits de la izquierda y los 32 bits de la derecha. Las 16 etapas restantes son funcionalmente idénticas, pero se parametrizan mediante diferentes funciones de la clave. El algoritmo se ha diseñado para permitir que el descifrado se haga con la misma clave que el cifrado, simplemente ejecutando los pasos en orden inverso. Cada una de las 16 etapas intermedias toma dos entradas de 32 bits y produce dos salidas de 32 bits. La salida de la izquierda es simplemente una copia de la entrada de la derecha. La salida de la derecha es el or exclusivo a nivel de bit de la entrada izquierda y una función de la entrada derecha y la clave de esta etapa K_i. Toda la complejidad reside en esta función.

La función consiste en cuatro pasos, ejecutados en secuencia. Primero se construye un número de 48 bits, E, expandiendo el R_{i-1} de 32 bits según una regla fija de trasposición y duplicación. Después, se aplica un or exclusivo a E y K_i. Esta salida entonces se divide en ocho grupos de 6 bits, cada uno de los cuales se alimenta a una caja S distinta. Cada una de las 64 entradas posibles a la caja S se transforma en una salida de 4 bits. Por último estos 8x4 bits se pasan a través de una caja P.

En cada una de las 16 iteraciones, se usa una clave diferente. Antes de iniciarse el algoritmo, se aplica una trasposición de 56 bits a la clave. Justo antes de cada iteración, la clave se divide en dos unidades de 28 bits, cada una de las cuales se gira hacia la izquierda una cantidad de bits dependiente del número de iteración. K_i se deriva de esta clave girada aplicándole otra trasposición de 56 bits. En cada vuelta se extrae y permuta de los 56 bits un subgrupo de 48 bits diferente.

8.4.2 Cifrado DES triple.

Como hemos visto, el algoritmo DES fue desarrollado inicialmente de forma que pudiera implementarse en un circuito electrónico con la tecnología de los años 70. Este hecho ha permitido que, con el avance de la electrónica digital, el algoritmo no solo pueda ser implementado en un circuito electrónico, sino que estos tengan la velocidad suficiente como para explorar, con un número reducido de circuitos electrónicos, todo el espacio de claves existente y descifrar un mensaje cifrado con el algoritmo DES en un tiempo razonable. En la actualidad, incluso los circuitos electrónicos pueden ser sustituidos por pruebas mediante ordenadores, no siendo necesario un gran número de ellos para poder descifrar el algoritmo en un tiempo de algunas semanas.

Por tanto, el cifrado con el algoritmo DES no es seguro, sin embargo, el cifrado triple con el algoritmo DES es valido. El método seleccionado, que se ha incorporado al estándar internacional 8732, se ilustra en la siguiente figura.

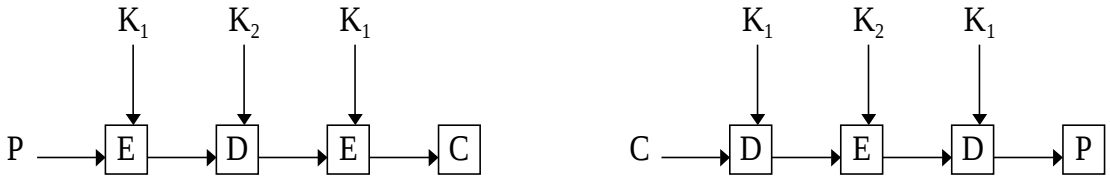


Figura 8.4.2.1: Esquema del cifrado DES triple.

Aquí se usan dos claves y tres etapas. En la primera etapa el texto normal se cifra con K_1. En la segunda etapa el algoritmo DES se ejecuta en modo de descifrado, usando K_2 como clave. Por último, se hace otro cifrado usando K_1. El hecho de que se usen dos claves y en modo EDE (cifrado-descifrado-cifrado) en lugar de EEE (cifrado-cifrado-cifrado) es debido a dos motivos:

1. En primer lugar, los criptógrafos admiten que 112 bits son suficientes para las aplicaciones comerciales por ahora. Subir a 168 bits (3 claves) simplemente agregaría carga extra innecesaria de administrar y transportar otra clave.

2. En segundo lugar, la razón de cifrar, descifrar y luego cifrar de nuevo es la compatibilidad con los sistemas DES de una sola clave. Tanto las funciones de cifrado como de descifrado son correspondencias entre números de 64 bits. Desde el punto de vista criptográfico, las dos correspondencias son igualmente robustas. Sin embargo, usando EDE en lugar de EEE, una computadora que usa cifrado triple puede hablar con otra que usa cifrado sencillo simplemente estableciendo $K_1=K_2$. Esta propiedad permite la introducción gradual del cifrado triple.

8.4.3 Cifrado IDEA (International Data Encryption Algorithm).

El algoritmo IDEA es un algoritmo de clave privada que fue diseñado por dos investigadores en Suiza, usa una clave de 128 bits, lo que lo hará inmune durante décadas a los ataques de la fuerza bruta. No hay ninguna técnica o máquina conocida actualmente que se crea que puede descifrar el algoritmo IDEA.

La estructura básica del algoritmo se asemeja al algoritmo DES en cuanto a que se alteran bloques de entrada de texto normal de 64 bits en una secuencia de iteraciones parametrizadas para producir bloques de salida de texto cifrado de 64 bits, como se puede ver en la figura siguiente.

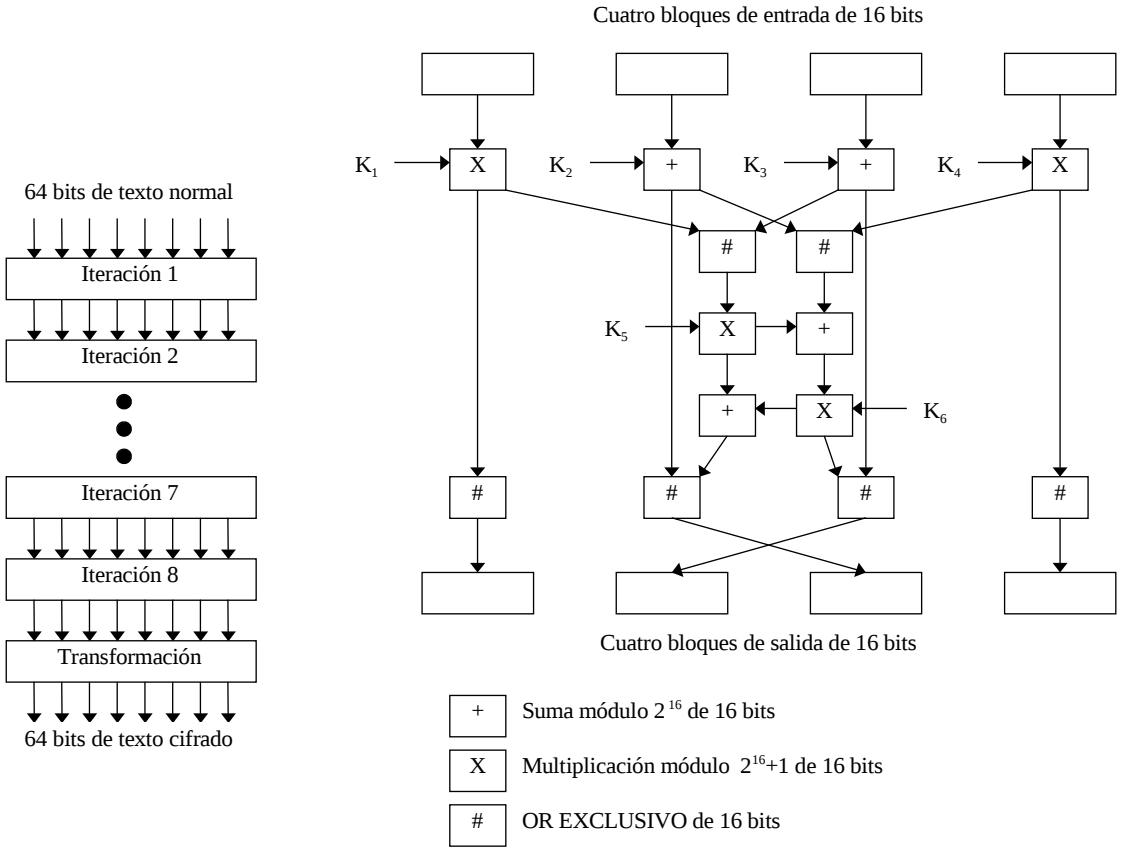


Figura 8.4.3.1: Algoritmo del cifrado IDEA.

Dada la extensa alteración de bits (por cada iteración, cada uno de los bits de salida depende de cada uno de los bits de entrada), basta con ocho iteraciones. Como con todos los cifrados de bloque, el algoritmo IDEA también puede usarse en el modo de realimentación de cifrado y en los demás modos del algoritmo DES. El algoritmo IDEA usa tres operaciones, todas sobre números sin signo de 16 bits. Estas operaciones son un or exclusivo, suma módulo 2^{16} y multiplicación módulo $2^{16}+1$. Las tres operaciones se pueden efectuar fácilmente en una microcomputadora de 16 bits ignorando las partes de orden mayor de los resultados. Las operaciones tienen la propiedad de que ningunos dos pares obedecen la ley asociativa ni la ley distributiva, dificultando el criptoanálisis. La clave de 128 bits se usa para generar 52 subclaves de 16 bits cada una, 6 por cada una de las ocho iteraciones y 4 para la transformación final. El descifrado usa el mismo algoritmo que el cifrado, sólo que con subclaves diferentes.

8.4.4 Cifrado AES (Advanced Encrytion Standard).

Después de comprobar la debilidad del algoritmo DES en su forma simple, diversos trabajos propusieron nuevos métodos de cifrados de bloques como BLOWFISH, Crab, FEAL, KHAFRE, LOKI91, NEWDES, REDOCII, SAFER K64 y el propio IDEA visto con anterioridad.

Sin embargo, en 1997 el National Institute of Standars and Technology propuso un concurso para escoger un nuevo algoritmo de cifrado. El cifrado debía cumplir los requisitos de ser de dominio público, utilizar criptografía de clave simétrica con bloques de 128 bits como mínimo, permitir claves de cifrado de 128, 192 y 256 bits y poder ser implementado tanto por software como por hardware. De todos los algoritmos presentados el algoritmo que resulto elegido fue el Rijndael, que ha pasado a ser conocido como AES.

AES opera en una matriz de 4x4 bytes en 10 rondas para una clave de 128 bits, en 12 rondas para una clave de 192 bits y en 14 rondas para una clave de 256 bits, más una ronda final. Cada una de las rondas consiste en una sustitución no lineal de cada byte de la matriz por otro de acuerdo a una tabla, una transposición donde las filas son rotadas de manera cíclica un número determinado de veces, un mezclado de unas columnas con otras utilizando una transformación lineal y una combinación final del resultado con la clave. La ronda final es similar a las anteriores solo que no se ejecuta el mezclado de las columnas con otras sino que se combinan dos veces los resultados anteriores con la clave.

El resultado de AES es lo suficientemente robusto como para que la NSA indicará en 2003 que AES en su versión de 128 bits podía utilizarse para cifrar información secreta y en sus versiones de 192 y 256 bits para cifrar información de alto secreto.

En la actualidad no se conoce ningún ataque, teórico o práctico sobre AES, pues algún ataque teórico, como el llamado ataque XSL, es objeto de controversia por los expertos en criptografía sobre si su planteamiento es matemáticamente correcto.

8.4.5 Cifrado RSA (Rivest, Shamir, Adleman).

Históricamente, el problema de distribución de claves siempre ha sido la parte débil de la mayoría de criptosistemas. Sin importar lo robusto que sea el criptosistema, si un intruso puede robar la clave, el sistema no vale nada.

En 1976, investigadores de la Universidad de Stanford propusieron una clase nueva de criptosistema en el que las claves de cifrado y descrifrado eran diferentes, y la clave de descifrado no podía derivarse de la clave de cifrado. En su propuesta, el algoritmo de cifrado (con clave), E, y el algoritmo de descifrado (con clave), D, tenían que cumplir los tres requisitos siguientes:

1. $D(E(P))=P$.

2. Es excesivamente difícil deducir D de E.

3. E no puede descifrarse mediante un ataque de texto normal seleccionado.

El método funciona como sigue. Una persona A que quiera recibir mensajes secretos diseña dos algoritmos E y D que cumplan los requisitos anteriores. El algoritmo de cifrado y la clave E_A de cifrado se hacen públicos, de ahí el nombre de criptografía de clave pública. Esto podría hacerse poniéndolos en un archivo accesible a

cualquiera que quiera leerlo. A publica también el algoritmo de descifrado, pero mantiene secreta la clave de descifrado D_A. Por tanto E_A es pública, pero D_A es secreta.

Ahora veamos si podemos resolver el problema de establecer un canal seguro entre A y B, que nunca han tenido contacto previo. Se supone que tanto la clave de cifrado de A, E_A, como la clave de cifrado de B, E_B, están en un archivo de lectura pública. Ahora, A toma su primer mensaje P, calcula $E_B(P)$ y lo envía a B. B entonces lo descifra aplicando su clave secreta D_B (es decir, calcula $D_B(E_B(P))=P$). Nadie más puede leer el mensaje cifrado $E_B(P)$ porque se supone que el sistema de cifrado es robusto y porque es demasiado difícil derivar D_B de la E_B públicamente conocida. A y B ahora pueden comunicarse con seguridad.

La única dificultad del método anterior estriba en que necesitamos encontrar algoritmos que realmente satisfagan los tres requisitos. Debido a las ventajas potenciales de la criptografía de clave pública, muchos investigadores han trabajando en este tema, y ya se conocen de forma pública algunos algoritmos. Un buen método fue descubierto por un grupo del M.I.T y es conocido como RSA por las iniciales de sus descubridores (Rivest, Shamir, Adleman). Su método se basa en:

1. Seleccionar dos números primos grandes, *p* y *q* (generalmente mayores que 10^{100}).

2. Calcular $n=pxq$ y $z=(p-1)x(q-1)$.

3. Seleccionar un número primo con respecto a *z* (es decir, un número sin ningún factor común con *z*), llamándolo *d*.

4. Encontrar e tal que $exd=1 \bmod z$

Con estos parámetros calculados por adelantado, estamos listos para comenzar el cifrado. Dividimos el texto normal (considerado como una cadena de bits) en bloques, para que cada mensaje de texto normal, P, caiga en el intervalo $0<P<n$, Esto puede hacerse agrupando el texto normal en bloques de k bits, donde k es el entero más grande para el que $2^k<n$ es verdad.

Para cifrar un mensaje, P, calculamos $C=P^e \pmod n$. Para descifrar C, calculamos $P=C^d \pmod n$. Puede demostrarse que, para todos los P del intervalo especificado, las funciones de cifrado y descifrado son inversas. Para ejecutar el cifrado, se necesitan e y n. Para llevar a cabo el descifrado se requieren d y n. Por tanto, la clave pública consiste en el par (e,n) y la clave privada consiste en (d,n).

La seguridad del método se basa en la dificultad de factorizar números grandes. Si el criptoanalista pudiera factorizar n (conocido públicamente), podría encontrar p y q, y a partir de éstos, z. Equipado con el conocimiento de z y de e, puede encontrar d usando el algoritmo de Euclides. Afortunadamente, los matemáticos han estado tratando de factorizar números grandes durante los últimos 300 años y las pruebas acumuladas sugieren que se trata de un problema excesivamente difícil. De acuerdo con los descubridores del RSA, la factorización de un número de 200 dígitos requiere 4 millones de años de tiempo de cómputo; la factorización de un número de 500 dígitos requiere 10^{22} años. En ambos casos se supone el mejor algoritmo conocido y una

computadora con un tiempo de instrucción de 1 nanosegundo. Aún si las computadoras continúan aumentando su velocidad, pasarán siglos antes de que sea factible la factorización de un número de 500 dígitos, y entonces simplemente se puede escoger un p y un q todavía más grandes.

Un ejemplo pedagógico trivial del algoritmo RSA es el siguiente, en el que queremos codificar las letras alfabeto castellano, representando la letra *a* por el valor 1, la letra *b* por el valor 2, etc.

Seleccionamos p=3 y q=11, dando n=33 y z=20. Un valor adecuado de d es d=7, puesto que 7 y 20 no tienen factores comunes. Con estas selecciones, e puede encontrarse resolviendo la ecuación 7e=1(mod 20), que produce e=3. El texto cifrado C, de un mensaje de texto normal, P, se da por la regla $C=P^3 \pmod{33}$. El texto cifrado lo descifra el receptor de acuerdo con la regla $P=C^7 \pmod{33}$.

Texto original (P)			Cifrado (C)	Texto descifrado (P)		
Carácter	Valor	P^3	$P^3 \pmod{33}$	C^7	$C^7 \pmod{33}$	Carácter
e	5	125	26	8031810176	5	e
n	14	2744	5	78125	14	n
r	19	6859	28	13492928512	19	r
i	9	729	3	2187	9	i
q	18	5832	24	4586471424	18	q
u	22	10648	22	2494357888	22	u
e	5	125	26	8031810176	5	e

Figura 8.4.4.1: Ejemplo de cifrado RSA.

Dado que los números primos escogidos para el ejemplo son tan pequeños, P debe ser menor que 33, por lo que cada bloque de texto normal puede contener sólo un carácter, con lo que el resultado es un cifrado por sustitución monoalfabética. En cambio, si hubiéramos seleccionado p y q del orden de 10^{100} podríamos tener n del orden de 10^{200}, por lo que cada bloque podría ser de hasta 664 bits (83 caracteres de 8 bits), contra 64 bits (8 caracteres de 8 bits) para el algoritmo DES. Sin embargo, el algoritmo RSA es demasiado lento para poder cifrar grandes volúmenes de datos, por lo cual suele usarse para distribuir claves de sesión de una sola vez para su uso con los algoritmos DES, IDEA u otros semejantes.

8.5 Validación de identificación en redes.

La validación de identificación es la técnica mediante la cual un proceso comprueba que su compañero de comunicación es quien se supone que es y no un impostor. La verificación de la identidad de un proceso remoto ante un intruso activo malicioso es sorprendentemente difícil y requiere protocolos complejos basados en criptografía.

El modelo general que usan todos los protocolos de validación de identificación es éste. Un usuario (en realidad, un proceso) iniciador, digamos Alfa, quiere establecer una conexión segura con un segundo usuario, Beta. Alfa y Beta se llaman principales, los personajes centrales de nuestra historia. Beta es un banquero con el que Alfa quiere hacer negocios. Alfa comienza por enviar un mensaje a Beta o a un centro de distribución de claves (KDC) confiable, que siempre es honesto. Siguen varios otros intercambios de mensajes en diferentes direcciones. A medida que se envían estos

mensajes, un intruso malicioso, Gamma, puede interceptarlos, modificarlos o reproducirlos para engañar a Alfa y a Beta, o simplemente para estropear sus actividades.

No obstante, una vez que se ha completado el protocolo, Alfa está segura de que está hablando con Beta, y Beta está seguro de que está hablando con Alfa. Es más, en la mayoría de los protocolos, los dos también habrán establecido una clave de sesión secreta para usarla durante toda la conversación subsiguiente. En la práctica, por razones de desempeño, todo el tráfico de datos se cifra usando criptografía de clave secreta, aunque la criptografía de clave pública se usa ampliamente con los protocolos de validación de identificación mismos y para establecer la clave de sesión.

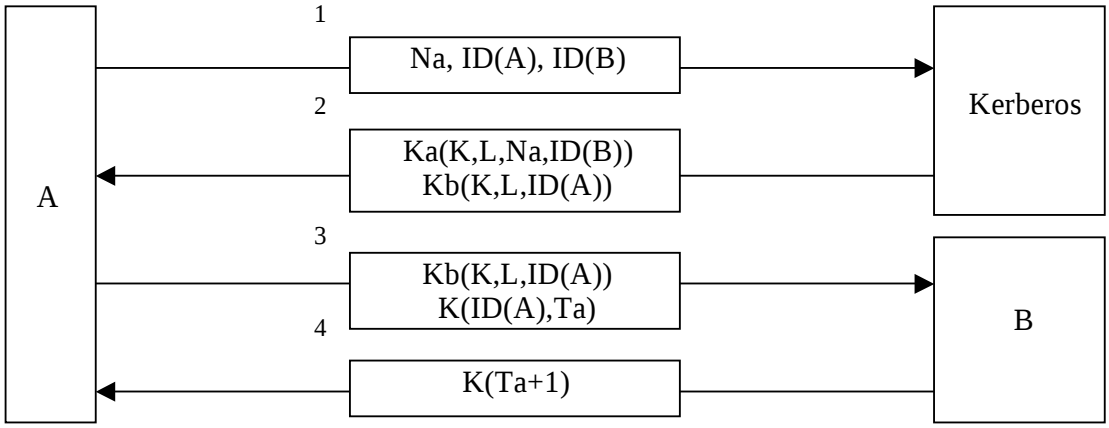
El objetivo es usar una clave de sesión nueva, seleccionada aleatoriamente, para cada nueva conexión es reducir al mínimo la cantidad de tráfico que es enviado con las claves secretas o públicas del usuario, reducir la cantidad de texto cifrado que puede obtener un intruso, y reducir al mínimo el daño provocado por la caída de un proceso, si su vaciado de memoria cae en las manos equivocadas. Con suerte, la única clave presente en ese momento será la clave de la sesión. Todas las claves permanentes debieron dejarse en blanco después de establecerse la sesión.

Existen diversos protocolos de validación que usan tanto claves secretas (Diffie-Hellman, Needham-Schroeder, Otway-Rees) como públicas (Rivest-Shamir). Sin embargo y dada su importancia, expondremos tan solo el protocolo de validación de clave secreta Kerberos.

8.5.1 Protocolo de autentificación Kerberos.

Kerberos es un protocolo de autenticación creado por el MIT que utiliza criptografía de claves simétricas para validar los usuarios o servicios ante otros usuarios o servicios de la red, evitando el envío de contraseñas sin cifrar a través de la misma, y con ello la posibilidad de que usuarios no autorizados puedan interceptar las contraseñas y acceder a los recursos sin autorización.

El funcionamiento básico del protocolo puede verse en la figura siguiente, en la cual solo consideramos los tres participantes básicos del mismo, el servidor de claves Kerberos y los dos usuarios y/o servicios (Alfa y Beta) que desean comunicarse.



Inicialmente, Alfa y Beta no comparten ninguna clave secreta y no pueden verificar la identidad del otro. Sin embargo, tanto Alfa como Beta comparten con Kerberos una clave criptográfica simétrica. Denominemos por $E_d(\ldots)$ al sistema criptográfico de clave simétrica *d* y sean *Ka* y *Kb* las claves secretas compartidas con Kerberos por Alfa y Beta respectivamente. Asimismo, sean *ID(A)* e *ID(B)* los respectivos identificadores de Alfa y Beta en la red de comunicación. Con estas especificaciones, el funcionamiento de Kerberos es:

1- Alfa solicita a Kerberos una credencial para identificarse ante Beta, así como una clave de sesión que le permita establecer dicha comunicación de forma segura. Para ello Alfa envía a Kerberos un valor aleatorio *Na* y tanto su identidad *ID(A)*, como la identidad de Beta con el que desea comunicarse, *ID(B)*.

2- Kerberos genera una clave de sesión aleatoria *K* y define el período de validez *L* de la clave generada, cifrando a continuación los valores *K*, *L*, *Na* y la identidad *ID(B)* de Beta con la clave secreta *Ka* de Alfa, obteniendo el valor *m* dado por $m = E_{Ka}(K, L, Na, ID(B))$. Además, Kerberos calcula una credencial *c* cifrando *K*, *L* y la identidad *ID(A)* de Alfa con la clave secreta *Kb* de Beta, es decir $c = E_{Kb}(K, L, ID(A))$, enviando la pareja de valores *(m, c)* obtenida a Alfa.

3- Alfa descifra *m* con su clave secreta *Ka* y recupera *K*, *L*, *Na* y la identidad *ID(B)* de Beta para el que fue emitida la credencial *c*. Entonces Alfa verifica que el valor aleatorio *Na* corresponde con el que él previamente envió y guarda *L* como referencia. En este punto, Alfa calcula la autenticación *a* para la credencial *c* cifrando su identidad *ID(A)* y un nuevo sello temporal *Ta* con la clave de sesión *K*, es decir $a = E_K(ID(A), Ta)$, enviando la pareja de valores *(c, a)* a Beta.

4- Beta recibe los valores *(c, a)* y descifra la credencial *c* con su clave secreta *Kb*, recuperando de esta forma *K*, *L* y la identidad *ID(A)*. Entonces utiliza la clave de sesión recuperada *K* para descifrar la autenticación y recuperar los valores *ID(A)* y *Ta* con el que comprueba que:

- Las identidades de la credencial y la autenticación coinciden.
- El sello temporal *Ta* es válido y se encuentra en los límites de *L*.

5- Si las comprobaciones son satisfactorias, Beta se convence de la autenticidad de la identidad de Alfa, así como que la clave de sesión cifrada en la credencial *c* es la misma con la que se ha cifrado la autenticación *a*. En tal caso, Beta envía a Alfa la conformidad *h* dada por $h = E_K(Ta+1)$.

6- Alfa descifra *h* con la clave de sesión *K* y verifica que el valor recuperado es *Ta+1*, lo cual asegura a Alfa que la clave de sesión *K* ha sido correctamente recibida por Beta.

Es conveniente hacer notar las diferentes funciones de los mensajes transmitidos en el protocolo anterior. Así por ejemplo, los valores cifrados *c* y *m* tienen la función de

garantizar la confidencialidad en la transmisión de la clave secreta de sesión *K*, mientras que los valores *a* y *h* se utilizan para la confirmación de la clave, es decir, para convencer a Alfa y Beta de que ambos poseen la misma clave secreta de sesión *K*.

El propósito del sello temporal *Ta* y del período de validez *L* es doble. Por una parte, tienen como objetivo que Alfa pueda utilizar la credencial c para realizar sucesivas autentificaciones ante Beta durante el período de validez de la clave sin necesidad de que intervenga Kerberos. El otro objetivo es el de prevenir un posible ataque activo al protocolo consistente en el almacenamiento de las claves para su posterior reutilización. Esto se consigue puesto que una clave no se acepta como válida si su sello temporal no se encuentra dentro de los límites del período de validez de la misma.

8.6 Resolución del control de integridad. Compendios de mensajes.

En muchas situaciones se requiere conocer la validez de un mensaje, esto es, conocer que no ha sido modificado por el camino y que nos llega tal y como fue escrito originalmente. A continuación describiremos un esquema de validación de identificación. Este esquema se basa en la idea de una función de dispersión unidireccional que toma una parte arbitrariamente grande de texto común y a partir de ella calcula una cadena de bits de longitud fija. Esta función de dispersión, llamada compendio de mensaje (message digest), tiene tres propiedades importantes:

1. Dado un texto P, es fácil calcular su compendio de mensaje MD(P).

2. Dado un compendio de mensaje MD(P), es imposible encontrar P.

3. Es imposible generar dos mensajes que sean correctos en un contexto y que tengan el mismo compendio de mensaje.

Para cumplir con el tercer criterio, la dispersión debe ser de cuando menos de 128 bits de longitud, y preferentemente mayor. Esta condición es necesaria pues el conjunto de posibles mensajes de entrada es prácticamente infinito (por no entrar en la inexactitud de decir que es infinito), mientras que el conjunto de compendios que se pueden generar es 2^n, siendo n la dispersión. Si la dispersión es muy pequeña, podrían generarse mensajes validos, por ejemplo un mensaje de felicitación a una persona y otro de insulto a la misma, que tuvieran el mismo compendio, y sustituir un mensaje por otro, con las evidentes consecuencias.

Los compendios de mensaje funcionan tanto en clave privada como en clave pública, siendo los de mayor uso el MD5 y el SHA.

8.6.1 Compendio de mensaje MD5.

El MD5 es la quinta de una serie de funciones de dispersión diseñadas por Ron Rivest. Opera alterando los bits de una manera tan complicada que cada bit de salida es modificado por cada bit de entrada. Su algoritmo es el siguiente:

- Se coge el mensaje original y se rellena hasta alcanzar una longitud de 448 módulo 512 bits, esto es, el mensaje debe tener una longitud en bits tal que al dividirla por 512 proporcione como resto de la operación el valor 448.

- A continuación se añade al mensaje la longitud original del mismo como un entero de 64 bits, por lo cual el mensaje total a codificar es un múltiplo de 512 bits.

- Se inicializa un buffer de 128 bits con un valor fijo.

- Ahora empieza el cálculo del compendio. Cada ronda toma un bloque de 512 bits de entrada y lo mezcla por completo con el buffer de 128 bits y los valores de una tabla construida a partir de la función matemática seno. Este proceso continúa hasta que todos los bloques de entrada se han consumido.

Una vez terminado el cálculo, el buffer de 128 bits contiene el valor del compendio de mensaje.

8.6.2 Compendio de mensaje SHA.

El SHA (Secure Hash Algoritm), fue desarrollado por la NSA y procesa los datos de entrada en bloques de 512 bits, pero a diferencia del MD5 genera un compendio de mensaje de 160 bits. Su algoritmo es el siguiente:

- Se coge el mensaje original y se rellena hasta alcanzar una longitud de 448 módulo 512 bits, esto es, el mensaje debe tener una longitud en bits tal que al dividirla por 512 proporcione como resto de la operación el valor 448.

- A continuación se añade al mensaje la longitud original del mismo como un entero de 64 bits, por lo cual el mensaje total a codificar es un múltiplo de 512 bits.

- Se inicializa un buffer de 160 bits con un valor fijo.

- Ahora empieza el cálculo del compendio. Cada ronda toma un bloque de 512 bits de entrada y lo mezcla con el buffer de 160 bits, utilizando 80 rondas para cada bloque de entrada y modificando cada 20 rondas las funciones de mezcla del bloque y el buffer. Este proceso continúa hasta que todos los bloques de entrada se han consumido.

Una vez terminado el cálculo, el buffer de 160 bits contiene el valor del compendio de mensaje. El SHA es 2^{32} veces más seguro que el MD5, pero es más lento su cálculo que el cálculo del MD5.

Hasta la fecha, tanto el MD5 como el SHA se han mostrado inviolables, pues aún con ataques sofisticados ideados, se tardarían más de 500 años en calcular los compendios de dos cartas con 64 variantes cada una, e incluso entonces no se garantiza una equivalencia.

8.7 Resolución del repudio: Firmas digitales.

La validación de identificación y autenticidad de muchos documentos legales, financieros y de otros tipos se determina por la presencia o ausencia de una firma manuscrita autorizada. Para que los sistemas computerizados de mensajes reemplacen el transporte físico de papel y tinta, debe encontrarse una solución a estos problemas.

El problema de inventar un reemplazo para las firmas manuscritas es difícil. Básicamente, lo que se requiere es un sistema mediante el cual una parte pueda enviar un mensaje "firmado" a otra parte de modo que:

1. El receptor pueda verificar la identidad proclamada del transmisor.

2. El transmisor no pueda repudiar después el contenido del mensaje.

3. El receptor no haya podido confeccionar el mensaje él mismo.

El primer requisito es necesario, por ejemplo, en los sistemas financieros. Cuando la computadora de un cliente ordena a la computadora de un banco que compre una tonelada de oro, la computadora del banco necesita asegurarse de que la computadora que da la orden realmente pertenece a la compañía a la que se le aplicará el débito.

El segundo requisito es necesario para proteger al banco contra fraudes. Supongamos que el banco compra una tonelada de oro, e inmediatamente después cae el precio del oro. Un cliente deshonesto podría demandar al banco, alegando que nunca emitió una orden para comprar el oro. Cuando el banco presenta el mensaje ante el juez, el cliente niega haberlo enviado.

El tercer requisito es necesario para proteger al cliente en el caso de que el precio del oro suba y que el banco trate de falsificar un mensaje firmado en el que el cliente solicitó un lingote de oro en lugar de una tonelada.

Al igual que la criptografía, las firmas digitales se dividen en dos grandes grupos, firmas de clave secreta y firmas de clave pública.

8.7.1 Firmas de clave secreta.

Un enfoque de las firmas digitales sería tener una autoridad central que sepa todo y en quien todos confíen, digamos X. Cada usuario escoge entonces una clave secreta y la lleva personalmente a las oficinas de X. Por tanto, sólo Alfa y X conocen la clave secreta de Alfa, K_A, etc.

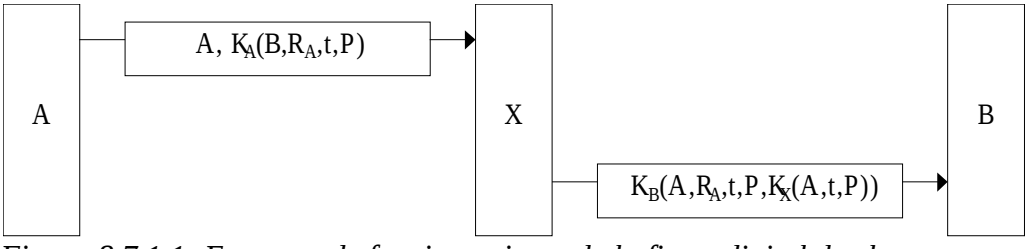


Figura 8.7.1.1: Esquema de funcionamiento de la firma digital de clase secreta.

Cuando Alfa quiere enviar un mensaje de texto normal firmado, P, a su banquero, Beta, genera $K_A(B,R_A,t,P)$ y lo envía como se muestra en la figura anterior. X ve que el mensaje es de Alfa, lo descifra y envía un mensaje a Beta como se muestra. El mensaje a Beta contiene el texto normal del mensaje de Alfa y también el mensaje firmado $K_X(A,t,P)$, donde t es una marca de tiempo. Ahora Beta atiende la solicitud de Alfa.

Si ahora Alfa niega el envío del mensaje, cuando el caso llega al juez y Alfa niegue haber enviado a Beta el mensaje, Beta indica que el mensaje vino de Alfa y no de un tercero Gamma pues X no hubiera aceptado un mensaje de Alfa a menos que estuviese cifrado con K_A, por lo que no hay posibilidad de que Gamma enviará a X un mensaje falso en nombre de Alfa. Beta además, presenta la prueba $K_X(A,t,P)$. Entonces el juez pide a X (en quien todo el mundo confía) que descifre la prueba. Cuando X testifica que Beta dice la verdad el caso queda resuelto.

Un problema potencial del protocolo de firma anterior es que Gamma repita cualquiera de los dos mensajes. Para minimizar este problema, se usan en todos los intercambios marcas de tiempo. Es más, Beta puede revisar todos los mensajes recientes para ver si se usó R_A en cualquiera de ellos. De ser así, el mensaje se descarta como repetición. Nótese que Beta rechazará los mensajes muy viejos con base en la marca de tiempo. Para protegerse contra ataques de repetición instantánea, Beta simplemente examina el R_A de cada mensaje de entrada para ver si un mensaje igual se recibió de Alfa durante el tiempo de validez de la marca temporal. Si no, Beta puede suponer con seguridad que ésta es una solicitud nueva.

8.7.2 Firmas de clave pública.

Un problema estructural del uso de la criptografía de clave secreta para las firmas digitales es que todos tienen que confiar en X. Es más, X lee todos los mensajes firmados. Los candidatos más lógicos para operar el servidor X son el gobierno, los bancos y los abogados. Estas organizaciones no tiene porqué inspirar confianza completa a todos los ciudadanos. Por tanto, sería bueno si la firma de documentos no requiriese una autoridad confiable.

Afortunadamente, la criptografía de clave pública puede hacer una contribución importante aquí. Supongamos que los algoritmos públicos de cifrado y descifrado tienen la propiedad de que $E(D(P))=P$ además de la propiedad normal de $D(E(P))=P$ (el RSA tiene esta propiedad por lo que el supuesto es razonable). Suponiendo que éste es el caso, Alfa puede enviar un mensaje de texto normal firmado, P, a Beta transmitiendo $E_B(D_A(P))$. Nótese que Alfa conoce su propia clave de descifrado (privada), D_A, así como la clave pública de Beta, E_B, por lo cual la construcción de este mensaje es algo que Alfa puede hacer.

Cuando Beta recibe el mensaje, lo transforma usando su clave privada, como es normal, produciendo $D_A(P)$, como se muestra en la figura siguiente:

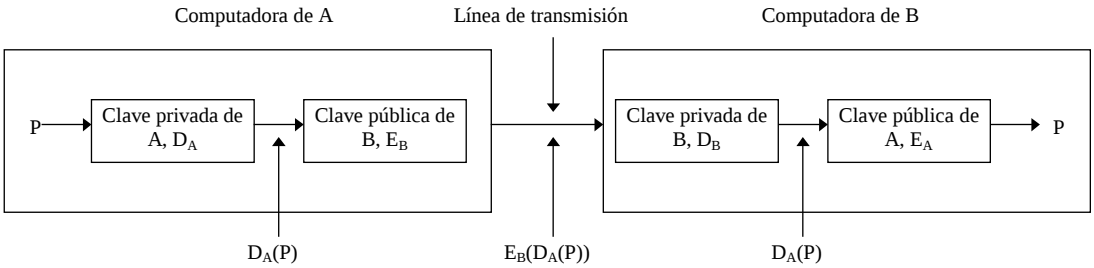


Figura 8.7.2.1: Esquema de funcionamiento de la firma digital de clave pública.

Beta almacena este texto en un lugar seguro y lo descifra usando E_A para obtener el texto normal original.

Para ver cómo funciona la propiedad de firma, supongamos que Alfa niega haber enviado el mensaje P a Beta. Cuando el caso llega al juez, Beta puede presentar tanto P como $D_A(P)$. El juez puede comprobar fácilmente que Beta tiene un mensaje válido cifrado por D_A con solo aplicarle E_A. Puesto que Beta no conoce la clave privada de Alfa, la única forma en que Beta pudo haber adquirido el mensaje cifrado con ella sería que Alfa en efecto lo hubiera enviado.

Sin embargo existen dos problemas, por un lado Beta puede demostrar que un mensaje fue enviado por Alfa siempre y cuando D_A permanezca en secreto. Si Alfa divulga su clave secreta, el argumento ya no se mantiene. Por otro lado, si Alfa decide cambiar su clave, algo legal y probablemente buena idea de forma periódica, cuando se aplique la actual E_A a $D_A(P)$ no se obtiene P. En consecuencia, parece que sí que se requiere alguna autoridad para registrar todos los cambios de clave y sus fechas.

En principio cualquier algoritmo de clave pública puede usarse para firmas digitales. El estándar de facto de la industria es el algoritmo RSA y muchos productos de seguridad lo usan.