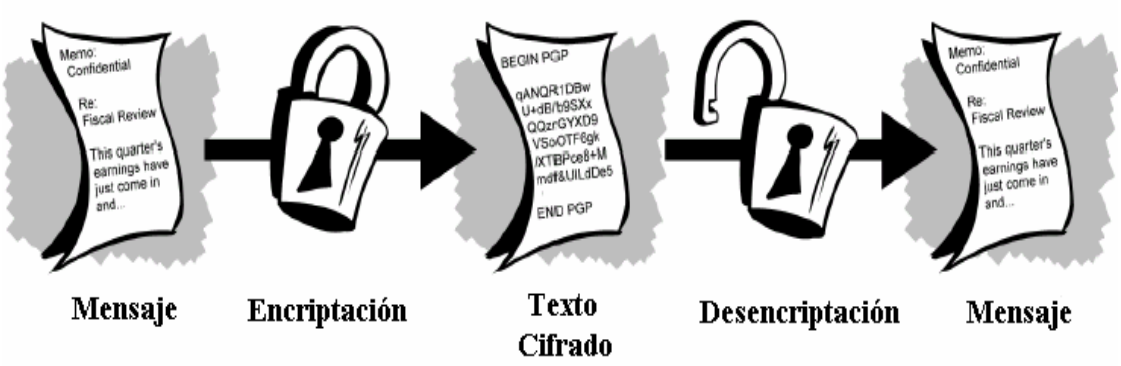
CAPITULO IV



ALGORITMOS DE ENCRIPTACIÓN, DESENCRIPTACIÓN - EVALUACIÓN Y VERIFICACIÓN

4.1.- Criptografía
4.2.- Algoritmos de Encriptación Simétricos
4.3.- Algoritmos de Encriptación Asimétricos
4.4.- Implementación de los algoritmos de Encriptación
4.5.- Evaluación de los Algoritmos a 64, 128 y 256 bits
4.6.- Análisis de resultados

Una red de comunicaciones es el conjunto de dispositivos hardware y software que permiten el intercambio de información digital entre los distintos elementos que se encuentran conectados en dicha red. El principal problema criptográfico es el de prevenir la extracción no autorizada de información sobre un canal inseguro y asegurar la privacidad. .

Existen varios problemas que los sistemas de seguridad deben evitar (interrupción, modificación, interceptación y usurpación) los cuales son resueltos por la criptografía. Los criptosistemas permiten establecer cuatros aspectos fundamentales de la seguridad informática: confidencialidad, integridad, autenticación y no repudio entre emisor y receptor.

Manejar diversos estándares de encriptado provee una mayor seguridad, además de que algunos protocolos de comunicación actuales permiten utilizar diferentes estándares de criptografía como son:

- TLS (*Transport Layer Security*))Encriptación a nivel TCP/IP
- SSL (*Secure Sockets Layer*) Encriptación a nivel TCP/IP
- EAP (*Extensible Authentication Protocol*) es una extensión del Protocolo punto a punto (PPP)
- SET *(Secure electronic transaction)*
- DSS *(Digital Satellite System)*

En este capitulo daremos una breve introducción a los algoritmos de encriptación como son RC4, DES, AES y RSA, también se realizara la evaluación y verificación de los mismos con encriptación de 64, 128 y 256 bits. Para entender la encriptación de los algoritmos daremos el concepto de criptografía.

4.1.- Criptografía

La Criptografía es el estudio de los códigos y claves que se emplean en la transmisión secreta de mensajes desde un emisor hasta un receptor como vemos en la Figura 4.1, el emisor y el receptor pueden ser una persona, un proceso en un sistema informático conectado a una red o un archivo almacenado en un disco de computadora o en otro dispositivo de almacenamiento.

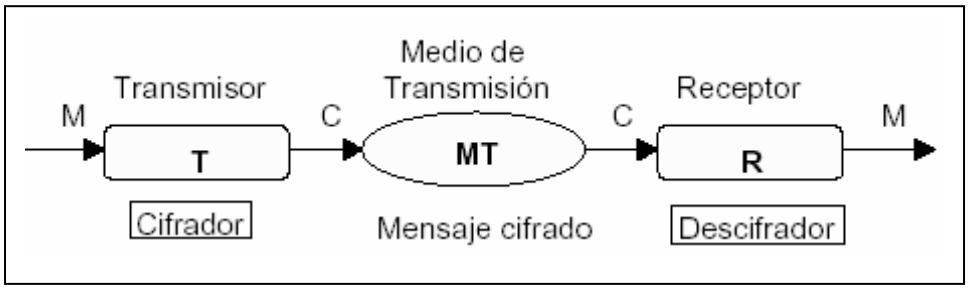


Figura 4.1: Sistema de cifrado Transmisor/Receptor

Los sistemas de cifrado son sistemas o métodos para tratar de hacer secreto los mensajes, y entre ellos se incluyen las claves de ocultación antes aludidas y los sistemas de cifrado por transposición y sustitución. Para nuestros fines, *cifrar* o *encriptar* es el proceso de transformar un texto conocido (*texto llano*) en una forma tremendamente complicada (*texto cifrado*), con el fin de que resulte ilegible para cualquiera que no posea la clave secreta. [LIB01].

El proceso inverso de reconstruir el texto original partiendo del texto cifrado y de una clave de descifrado es el *descifrado* o *desencriptación***.**

ABCDEFGHIJKL	*(texto llano o original)*
#@2€%223?¬€¿¡”	*(texto cifrado o encriptado)*

El cifrado se puede entender como el hecho de guardar algo valioso dentro de una caja fuerte cerrada con llave. Los datos confidenciales se cifran con un algoritmo de clave que los deja ilegibles si no se conoce dicha clave. Las claves de cifrado de datos se determinan en el momento de la conexión, entre la conexión y el equipo

del otro extremo. El uso del cifrado de datos puede iniciarse en su equipo o en el servidor al que se conecta.

Los algoritmos de encriptación están formados por encriptación simétrica y asimétrica, los cuales utilizan claves públicas y privadas respectivamente para la protección de los datos.

4.1.1.- Encriptación Simétrica

En este tipo de criptografía tanto el emisor como el receptor utilizan la misma clave, como vemos en la Figura 4.2. El conocimiento de la clave implica el poder encriptar y desencriptar mensajes. En la encriptación simétrica normalmente se utilizan dos claves: una para encriptar y otra para desencriptar.

Normalmente se emplea una clave ya que conociendo la clave de encriptación es fácil, la mayoría de las veces inmediato, calcular la clave de desencriptación y viceversa. [www17].

Ventajas:

- Son algoritmos “fáciles” de implementar.
- Requieren “poco” tiempo de cómputo.

Desventajas:
Las claves han de transmitirse por un canal seguro. Algo difícil de realizar.

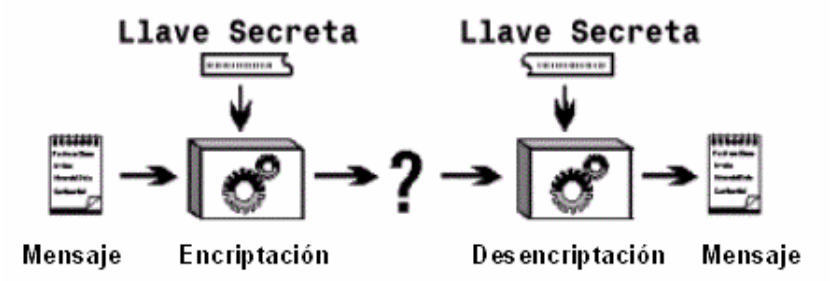


Figura 4.2: Criptografía Simétrica

Cuando la clave de cifrado es la misma que la de descifrado (y, por tanto, los algoritmos de cifrado y de descifrado coinciden) se habla de algoritmo de cifrado simétrico. Este es el tipo de cifrado que ha dominado la historia de la criptografía hasta hace un par de décadas. Existen muchos algoritmos de clave simétrica resistentes al criptoanálisis: DES, RC4, AES, IDEA, CAST, Triple-DES.

4.1.2.- Encriptación Asimétrica

La criptografía asimétrica surge para solucionar el problema que tiene la criptografía simétrica de distribución de la clave. Una solución a esto la dieron "*Diffie*" y "*Hellman*" en 1976. Su solución proponía utilizar "*funciones de un sentido*" en canales abiertos, como veremos:

La función f(*x*) en un sentido, entonces:

- Es fácil calcular y = f(x), conocido x
- Conociendo y computacionalmente es imposible el calculo de $x = f^{-1}(y)$

Los cifrados asimétricos son mucho más flexibles desde el punto de vista de la administración de claves. Cada usuario tiene un par de claves: una clave pública y una clave privada. Los mensajes cifrados con una clave pública pueden ser descifrados solamente por la clave privada, ver Figura 4.3. La clave pública puede ser ampliamente diseminada, mientras que la clave privada se mantiene en secreto [www17].

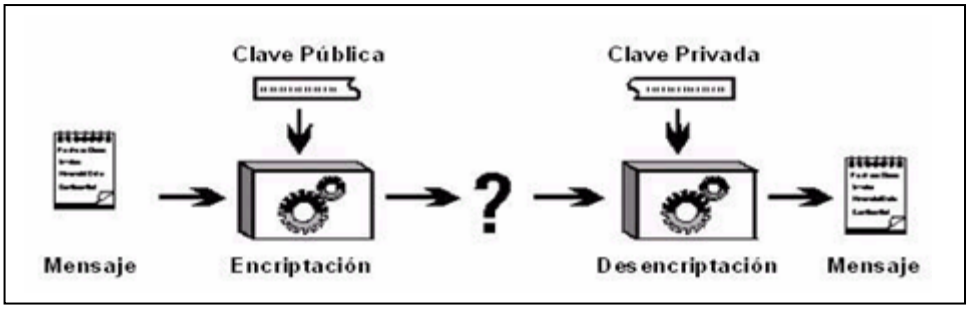


Figura 4.3: Criptografía Asimétrica

Supongamos que tanto Ana como Belén tienen un par de claves pública/privada. Si Ana desea enviar un mensaje a Belén, los pasos a seguir son los siguientes:

- Ana obtiene la clave pública de Belén *k*
- Ana compone un mensaje *M* y lo cifra con la clave pública de Belén.
- Ana envía el mensaje cifrado *Ck(M)*
- Belén recibe el mensaje y le aplica su clave privada *k'* obteniendo Dk'(Ck(M)) = M

Es decir, cualquiera puede cifrar un mensaje y enviárselo a Belén, pero solamente ella podrá descifrar los mensajes que le llegan, nadie más, ni siquiera Ana podrá descifrar el mensaje que acaba de cifrar. En la actualidad, los sistemas o algoritmos de clave pública más utilizados son el RSA y el Diffie-Hellman.

4.2.- Algoritmos de encriptación Simétricos

Son sistemas convencionales basados en password (clave) que la mayoría de la gente conoce. El usuario suministra una clave y el archivo es cifrado con la clave. Para descifrar el archivo, el usuario debe suministrar la misma clave nuevamente y el proceso es reversado. La clave es la llave de encriptación [www18]. El principal problema aquí es la clave; el emisor y el receptor no solamente deben de estar de acuerdo en usar la misma clave, sino que también deben idear alguna manera para intercambiarla, especialmente si están en diferentes áreas geográficas. En los sistemas de clave privada, la integridad de la clave es sumamente importante. Por lo tanto, es importante reemplazar periódicamente esta clave.

En este punto es necesarios aclarar que existen dos técnicas de cifrado simétrico, las cuales son: algoritmos de cifrado en bloque[1] y algoritmos de cifrado en flujo[2].

[1] Sistema criptográfico que cifra de bloque en bloque, usualmente cada bloque es de 128 bits
[2] Sistema criptográfico que cifra de bit en bit.

Los algoritmos criptográficos simétricos (RC4, DES, AES) serán analizados y evaluados para obtener resultados finales.

4.2.1.- Algoritmo RC4

El algoritmo de encriptación RC4 fue diseñado por Ron Rivest de la RSA Security en el año 1987; su nombre completo es Rivest Cipher 4 teniendo el acrónimo RC un significado alternativo al de Ron's Code utilizado para los algoritmos de cifrado RC2, RC5 y RC6.

RC4 o ARC4 es parte de los protocolos de encriptación más comunes como WEP, WPA para dispositivos wireless y TLS. Utilizando 40 y 128 bits correspondientes a la clave secreta de encriptación y los 24 y 48 bits al vector de inicialización, por lo tanto WEP utiliza los 64 bits y WPA los 128 bits.

El algoritmo RC4 utiliza un vector de inicialización (VI), este es generado dinámicamente y debería ser diferente para cada trama. El objetivo perseguido con el Vector de Inicalización es cifrar con claves diferentes para impedir que un posible atacante pueda capturar suficiente tráfico cifrado con la misma clave y terminar finalmente deduciendo la clave. Como es lógico, ambos extremos deben conocer tanto la clave secreta como el Vector de Inicialización. Lo primero sabemos ya que es conocido puesto que está almacenado en la configuración de cada elemento de red. El Vector de Inicalización, en cambio, se genera en un extremo y se envía en la propia trama al otro extremo, por lo que también será conocido.

Observemos que al viajar el Vector de Incialización en cada trama es sencillo de interceptar por un posible atacante. Debido a las vulnerabilidades que tiene el algoritmo RC4 fue excluido de los estándares de alta seguridad por los criptógrafos, algunos modos de usar el algoritmo de criptografía RC4 lo han llevado a ser un sistema de criptografía muy inseguro, incluyendo su uso WEP. No está recomendado su uso en los nuevos sistemas, sin embargo, algunos sistemas basados en RC4 son lo suficientemente seguros para un uso común [www19].

Pasos para la encriptación

1. Se calcula un CRC de 32 bits de los datos. Este CRC-32 es el método que propone WEP para garantizar la integridad de los mensajes (ICV, *Integrity Check Value*).
2. Se concatena la clave secreta a continuación del IV formado el *seed.*
3. El PRNG (*Pseudo-Random Number Generator*) de RC4 genera una secuencia de caracteres pseudoaleatorios (*keystream = VI + Key (clave)*), a partir del *seed*, de la misma longitud que los bits obtenidos en el punto 1.
4. Se calcula la O exclusiva (XOR) de los caracteres del punto 1 con los del punto 3. El resultado es el mensaje cifrado.
5. Se envía el IV (sin cifrar) y el mensaje cifrado dentro del campo de datos (*frame body*) de la trama IEEE 802.11.

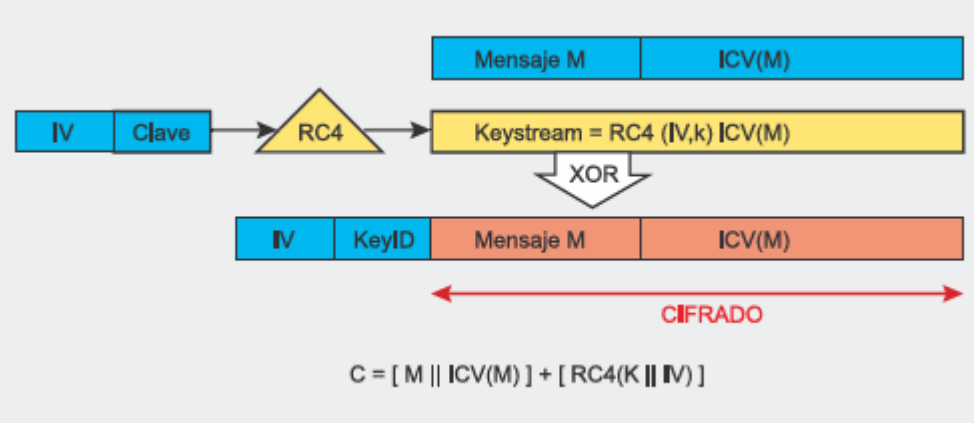


Figura 4.4: Estructura del WEP [www25]:

4.2.2.- Algoritmo DES

El estándar de encriptación de datos DES (Data Encyption Standard), adoptado en 1977 por el Instituto Nacional de Estándares y Tecnología NIST[3] **,** es el estándar Federal de procesamiento de la información. El esquema global del encriptado DES se muestra en la Figura 4.5, como con cualquier esquema de encriptado y la clave. En este caso, el texto nativo debe tener una longitud de 64 bits y la clave 56 bits.

[3] **NIST**: Nacional Institute of Standards and Technolgy.

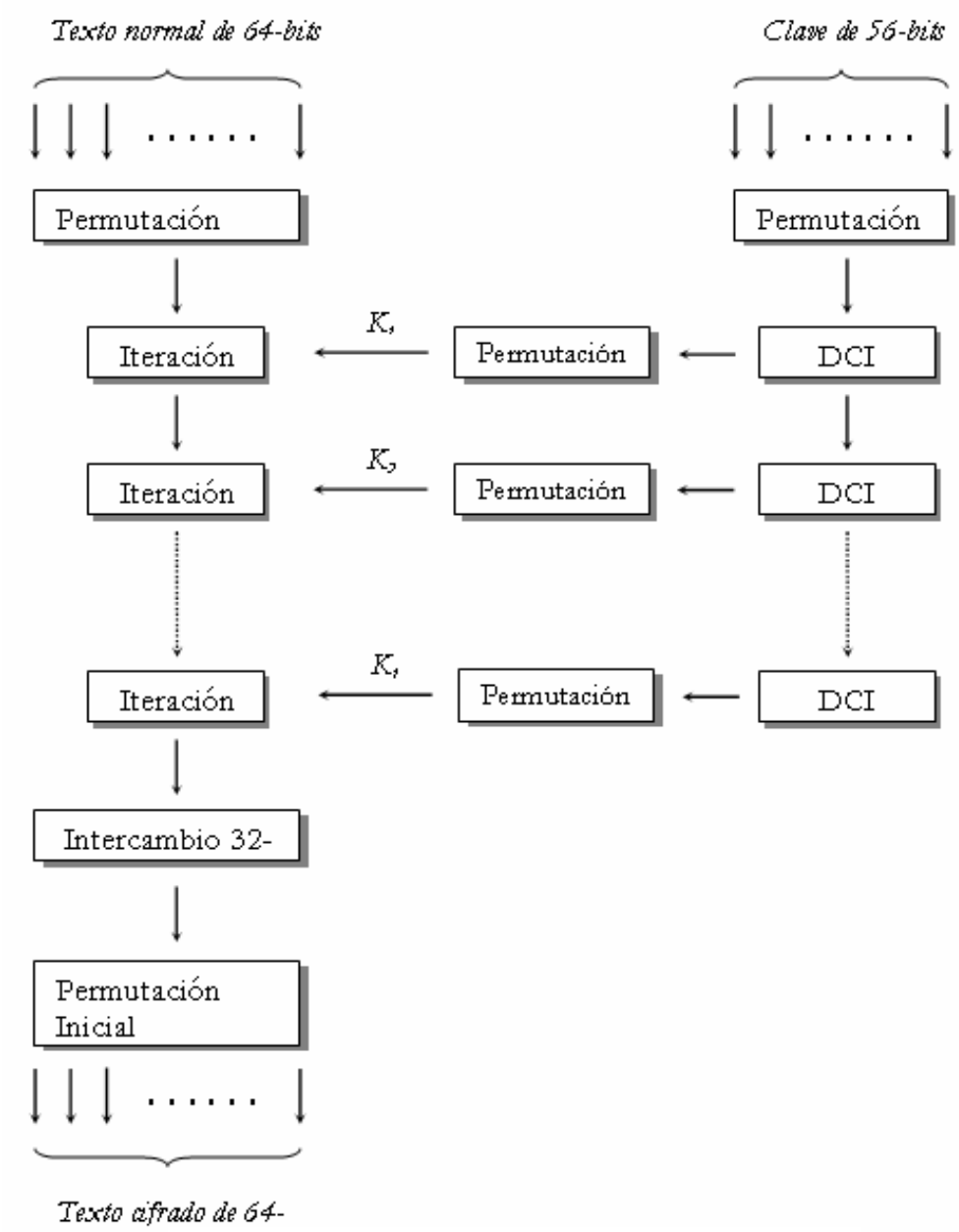


Figura 4.5 Esquema general del algoritmo DES [LIB03].

En la parte derecha de la Figura 4.5**,** muestra la forma como se usan los 56 bits de la clave, inicialmente, la clave se transforma por una función de permutación. Después, para cada una de las 16 interacciones, se produce una subclave (Ki) por medio de un desplazamiento circular y una permutación. La función de permutación es la misma para cada iteración, pero se produce una subclave diferente debido al desplazamiento repetido de los bits de la clave [LIB03].

Las 16 etapas siguientes denominadas *iteraciones* son funcionalmente idénticas, pero cada una se parametriza con una parte diferente de la clave. La penúltima función "*Intercambio 32-bits"* como su nombre lo indica, intercambia los 32 bits de la izquierda y los 32 bits de la derecha. La última función es el inverso exacto de la permutación inicial.

La función de permutación inicial con un bloque de entrada de 64 bits, se permuta de la siguiente manera:

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Es decir, la entrada permutada tiene como primer bit el bit 58 del original, como segundo bit el bit 50 del original, y así sucesivamente, hasta llegar al último bit, que corresponderá al bit 7 del texto sin cifrar. La función de permutación inicial inversa se reordena de la siguiente manera:

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Es decir, la entrada permutada inversa tiene como primer bit el bit 40 de la salida de la función intercambio 32-bits, como segundo bit el bit 8 de la salida de la función intercambio 32-bits, y así sucesivamente, hasta llegar al último bit, que corresponderá al bit 25 de la salida de la función intercambio 32-bits.

Los 56 bits de la clave son usados de la siguiente manera: Inicialmente la clave se reordena por medio de una función de permutación. A continuación, para cada una de las iteraciones, se genera una subclave K_i por medio de la función DCI

(desplazamiento circular a la izquierda) y otra función de permutación que es la misma para cada iteración. Note que se produce una subclave diferente para cada iteración debido al desplazamiento repetido de los bits de la clave suministrada.

En la Figura 4.6 se muestra el procesamiento para una iteración del algoritmo DES, el cual se puede resumir en las siguientes fórmulas:

$L_i = R_{i-1}$

$R_i = L_{i-1} \ominus f(R_{i-1}, K_i)$ donde ⊖ denota la función XOR bit a bit.

La parte izquierda L_i de la salida de una iteración es igual a la parte derecha R_{i-1} de la entrada a esa iteración. La parte derecha de la salida R_i es la operación XOR de L_{i-1} y una función compleja de R_{i-1} y K_i. La *"función compleja"* lleva a cabo cuatro pasos sobre la salida derecha, mediante una transposición basada en la operación OR-Exclusivo.

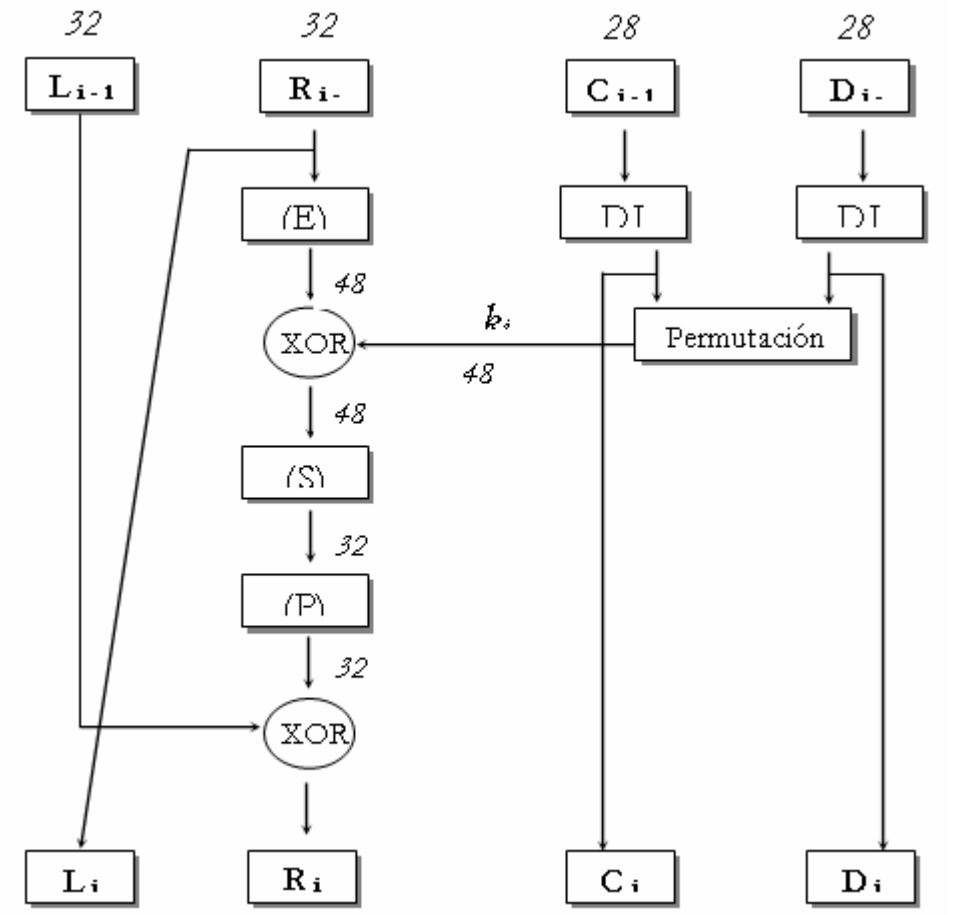


Figura 4.6 Iteración del Algoritmo DES [LIB03]

1. La mitad derecha R_{i-1} de 32 bits se convierte, mediante una regla de expansión y permutación, en el número *E*, de 48 bits, los mismos que se reordenan de la siguiente manera:

32	1	2	3	4	5	
	4	5	6	7	8	9
8	9	10	11	12	13	
	12	13	14	15	16	17
16	17	18	19	20	21	
	20	21	22	23	24	25
24	25	26	27	28	29	
	28	29	30	31	32	1

2. *E* y *K* se combinan mediante un OR-Exclusivo.
3. Los 48 bits generados en la etapa 2 se dividen en ocho grupos de 6 bits que se introducen en las cajas "*S*", cada una de las cuales produce 4 bits de salida, es decir, simplemente traducen cada combinación de entrada de 48 bits en un modelo particular de 32 bits.
4. Los 32 bits restantes se introducen en la caja de permutación *P* que consiste en el siguiente reordenamiento:

	16	7	20	21
29	12	28	17	
	1	15	23	26
	5	18	31	10
	2	8	24	14
	32	27	3	9
	19	13	30	6
	22	11	4	25

La clave de 56 bits se trata como dos cantidades de 28 bits, rotuladas como C_{i-1} y D_{i-1}. En cada iteración estas cantidades sufren de forma separada un desplazamiento circular a la izquierda (DI), los cuales sirven como entrada para la siguiente iteración y también como entrada a otra función de permutación que produce una salida de 48 bits que sirve como entrada a la función $f(R_{i-1}, K_i)$. Los desplazamientos a la izquierda son de dos bits, salvo para las rondas 1, 2, 9 y 16, en las que se desplaza sólo un bit.

El proceso de descifrado del DES es básicamente el mismo que el proceso de cifrado con una pequeña variación. La idea es, usar el texto cifrado como entrada al algoritmo DES, pero usar la clave en orden inverso, es decir, utilizar la subclave K_{16} en la primera iteración, la subclave K_{15} en la segunda iteración y así sucesivamente hasta utilizar la subclave K_1 en la última iteración.
Este algoritmo ha sido motivo de gran controversia, parte de ella se debe al secreto que rodeó a su desarrollo. IBM trabajó en colaboración con la Agencia Nacional de Seguridad de Estados Unidos y ambas guardaron el secreto de los aspectos del diseño del algoritmo. Muchas de las críticas obtenidas se encuentran en el hecho de usar sólo 56 bits de parte de la clave para conseguir el cifrado, esto ya es considerado por muchos insuficientes.

Evidentemente para romper una clave semejante sería necesaria una enorme cantidad de potencia de cálculo. Sin embargo, no es una tarea imposible. Los ordenadores de alta velocidad, mediante análisis estadísticos, no necesitan emplear todas las posibles combinaciones para romper la clave. A pesar de ello, el objetivo de DES no es proporcionar una seguridad absoluta, sino únicamente un nivel de seguridad razonable para las redes orientadas a aplicaciones comerciales.

Una variante al DES es usar el Triple DES. Consiste en aplicar varias veces el algoritmo DES con diferentes claves al mensaje original. Responde a la siguiente estructura:

$$C = E_{k1} (D_{k2} (E_{k1} (M)))$$

Es decir, codificamos con la subclave k_1, decodificamos con k_2 y volvemos a codificar con k_1. La clave resultante es la concatenación de k_1 y k_2, con una longitud de 112 bits. La razón de que se usan sólo dos claves en lugar de tres es que una clave de 112 bits es suficiente para las aplicaciones comerciales por ahora. Subir a 168 bits simplemente agregaría una carga extra innecesaria al proceso de cifrado y descifrado.

4.2.3.- Algoritmo AES

En 1998, investigadores de doce países diferentes propusieron quince candidatos para el estándar AES (Advanced Encryption Standard, estándar avanzado de cifrado), el nuevo método de cifrado (y sucesor de DES/3DES), adoptado por el Gobierno Federal de los Estados Unidos. AES es un algoritmo público diseñado para proteger información gubernamental durante los primeros años de este siglo.

NIST solicitó propuestas para el estándar AES el 12 de septiembre de 1997, cada uno de los algoritmos candidatos soporta tamaño de clave criptográfica de 128, 192 y 256 bits. Con un tamaño de clave de 128 bits hay aproximadamente 340.000.000.000.000.000.000.000.000.000.000.000.000 (340 seguido de 36 ceros posibles claves). AES también conocido como Rijndael es un sistema de cifrado de bloques diseñado por Joan Daemen y Vincent Rijmen.

El sistema de cifrado tiene una longitud de bloque y una longitud de clave variables. Actualmente, está especificando cómo utilizar claves con una longitud de 128, 192 o 256 bits para cifrar bloques con longitudes de 128, 192 o 256 bits, pudiéndose utilizar las nueve combinaciones posibles.La longitud de clave como la de bloque pueden extenderse en múltiplos de 32 bits. Rijndael puede implementarse en un amplio rango de procesadores y plataformas. El diseño del algoritmo Rijndael fue influenciado por el diseño del sistema de cifrado de bloques.

El problema real de distribución de claves para un gran grupo de usuarios que deseen comunicarse en función del número de usuarios: se requiere ½ $(n^{2}-n)$ claves para un grupo de n usuarios. La Tabla 4.1 muestra una serie de ejemplos del número de usuarios y del número de claves requeridas, ½ $(n^{2}-n)$, correspondiente. Por ejemplo para dotar de seguridad a una gran corporación, a un servicio postal o al ejército de un país, es preciso resolver un importante problema de distribución de claves secretas [LIB01].

n	**½ (n^2-n)**
2	1
3	3
10	45
100	4.9500
1000	499.995.000

Tabla 4.1: Número de usuarios, n, y número de claves necesarias correspondiente, ½ (n^2-n) [LJB01].

4.3.- Algoritmo de Encriptación Asimétrica

También se los conoce como algoritmos de clave pública. Cada parte obtiene un par de claves, una pública y una privada. La clave pública esta hecha para que todos la conozcan mientras que la privada no La ventaja de utilizar criptografía de clave pública es la seguridad y la conveniencia. La clave privada nunca necesita transmitirse o confiarse a alguien más. Por lo tanto no hay ninguna posibilidad de que la clave se comprometa ni de que la transmisión sea interceptada o decodificada. [www18].

Es necesario aclarar que la clave privada utilizada en estos algoritmos no es la misma clave utilizada en los criptosistemas de clave privada; la clave privada sólo descifra los mensajes que han sido cifrados con la clave pública asociada.

Entre los algoritmos más representativos de esta categoría son el algoritmo Diffie-Hellman y el algoritmo RSA.

4.3.1.- Algoritmo RSA

El algoritmo RSA (Rivest, Shamir, Adleman) es un algoritmo de clave pública desarrollado en 1977 en el MIT por Ronald Rivest, Adi Shamir y Leonard Adelman. Fue registrado el 20 de Septiembre de 1983. El 20 de Septiembre del 2000, tras 17 años, expiró la patente RSA, pasando a ser un algoritmo de dominio público. Este popular sistema se basa en el problema matemático de la factorización de números grandes.
El algoritmo RSA funciona de la siguiente manera:

1. Inicialmente es necesario generar aleatoriamente dos números primos grandes, a los que llamaremos p y q.
2. A continuación calcularemos n como producto de p y q: n = p * q
3. Se calcula fi: fi(n)=(p-1)(q-1)
4. Se calcula un número natural e de manera que MCD(e, fi(n))=1 , es decir e debe ser primo relativo de fi(n). Es lo mismo que buscar un numero impar por el que dividir fi(n) que de cero como resto.
5. Mediante el algoritmo extendido de Euclides se calcula d: e.d mod fi(n)=1 Puede calcularse d=((Y*fi(n))+1)/e para Y=1,2,3,... hasta encontrar un d entero.
6. El par de números (e,n) son la clave pública.
7. El par de números (d,n) son la clave privada.
8. Cifrado: La función de cifrado es C = M^e mod n
9. Descifrado: La función de descifrado es M = C^d mod n

Ejemplo con números pequeños.

1. Escogemos dos números primos, por ejemplo p=3 y q=11.
2. n = 3 * 11 = 33.
3. fi(n) = (3-1) * (11-1) = 20.
4. Buscamos e: 20/1=0, 20/3=6.67. e=3.
5. Calculamos d como el inverso multiplicativo módulo z de e, por ejemplo, sustituyendo Y por 1,2,3,... hasta que se obtenga un valor entero en la expresión: d = ((Y * fi(n)) + 1) / e = (Y * 20 + 1) / 3 = 21 / 3 = 7
6. e=3 y n=33 son la clave pública.
7. d=7 y n=33 son la clave privada.
8. Cifrado: Mensaje = 5, C = M^e mod n = 5^3 mod 33 = 26
9. Descifrado: M = C^d mod n = 26^7 mod 33 = 8031810176 mod 33 = 5

4.4.- Implementación de los algoritmos de encriptación

Si bien existen los algoritmos de encriptación su implementación sería poco práctica si consideramos que las herramientas actuales de desarrollo ya disponen de librerías que nos permiten encriptar y desencriptar información por cualquiera de los algoritmos hasta ahora conocidos.

La herramienta de desarrollo que en mayor medida tiene procedimientos de encriptado es Visual Studio 2005 de Microsoft, a través de las clases contenidas en el la librería "`Cryptography`", esta librería nos da la disponibilidad de usar los siguientes algoritmos de encriptacion: DES, TripleDES, Rijndael, entre otros; de esta forma se permiten desarrollar aplicaciones con encriptados seguros e incluso combinarles tipos de encriptacion.

4.5.- Evaluación de algoritmos a 64, 128 y 256 bits

Para le evaluación de los algoritmos de encriptación se procedió a desarrollar un software en Visual Studio 2005, mismo que encripta un archivo con los métodos DES, Rijndel y RC4.

El proceso de este software es ubicar un archivo encriptarlo y desecriptarlo para ubicar cual de ellos es el más rápido, además cambiar las claves de 64,128 y 256 bits para verificar cual de ellos sería mas seguro a nivel de la clave.

En lo cual hay que considerar que el DES únicamente trabaja a 64 bits, por lo que será evaluado la velocidad de encriptación y desecriptación en comparación con los demás.

Las pruebas a desarrollar son las siguientes:

	Tamaño de Archivo	Tamaño de Clave en bits	Método de Encriptación
Test 1	500k	64	DES RC4 RIJNDEL (AES)
Test 2	500k	128	RC4 RIJNDEL (AES)
Test 3	500k	256	RC4 RIJNDEL (AES)
Test 4	1m	64	DES RC4 RIJNDEL (AES)
Test 5	1m	128	RC4 RIJNDEL (AES)
Test 6	1m	256	RC4 RIJNDEL (AES)
Test 7	10m	64	DES RC4 RIJNDEL (AES)
Test 8	10m	128	RC4 RIJNDEL (AES)
Test 9	10m	256	RC4 RIJNDEL (AES)

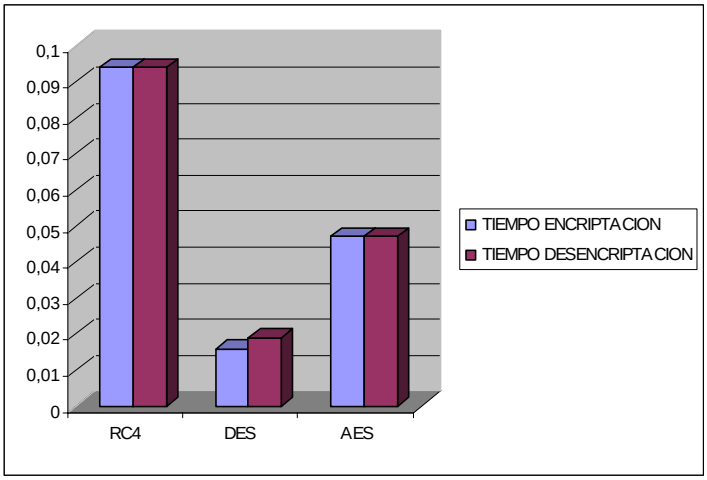
Resultados de las Pruebas:

4.5.1.- TEST 1

Tamaño de Archivo: **500k**
Tamaño de Clave: **64 bits**

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,094	0,094
DES	0,016	0,019
AES	0,047	0,047

Tabla 4.2: Encriptación a 64 Bits



El resultado que se obtuvo con clave de 64 bits y los tiempos mínimos fue:

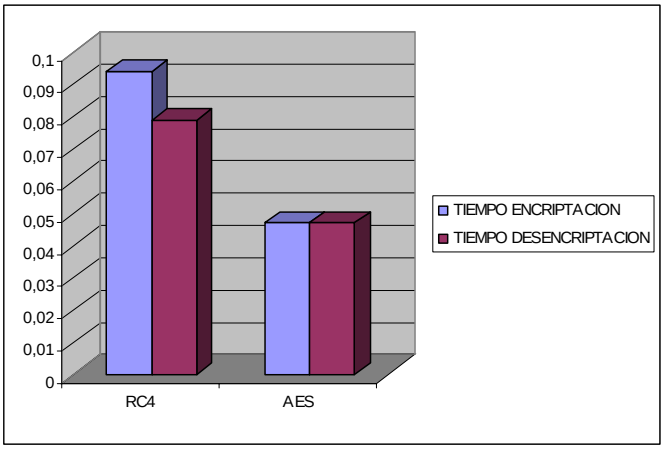
Encriptación: Algoritmo DES con 0,016 Bits

Desencriptación: Algoritmo DES con 0,019 Bits

Tamaño de Clave 128 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,094	0,079
AES	0,047	0,047

Tabla 4.3: Encriptación de 128 Bits



El resultado que se obtuvo con clave de 128 bits y los tiempos mínimos fue:

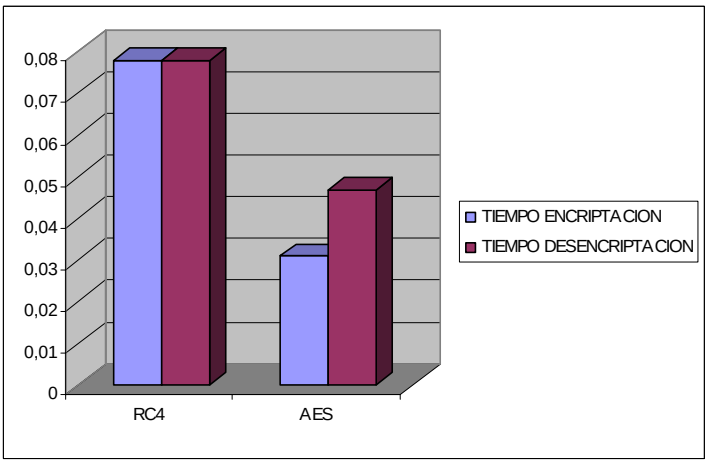
Encriptación: Algoritmo AES con 0,047 Bits

Desencriptación: Algoritmo AES con 0,047 Bits

Tamaño de Clave 256 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,078	0,078
AES	0,031	0,047

Tabla 4.4: Encriptación de 256 Bits



El resultado que se obtuvo con clave de 256 bits y los tiempos mínimos fue:

Encriptación: Algoritmo AES con 0,031 Bits

Desencriptación: Algoritmo AES con 0,047 Bits

4.5.2.- TEST 2

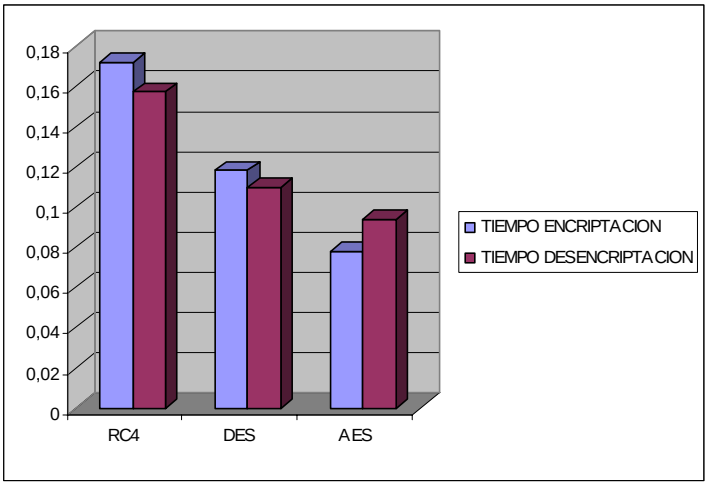
Tamaño de Archivo: **1 Mb**
Tamaño de Clave: **64 bits**

En este Test se trabajo con un archivo de tamaño 1 Mb, algoritmos RC4, DES y AES con clave de 64 bits, algoritmos RC4 y AES con claves de 128 y 256 bits, y sus tiempos de encriptación y desencriptación. Como se ve en las siguientes tablas y datos estadísticos

Tamaño de Clave 64 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,172	0,157
DES	0,118	0,11
AES	0,078	0,094

Tabla 4.5: Encriptación a 64 Bits



El resultado que se obtuvo con clave de 64 bits y los tiempos mínimos fue:

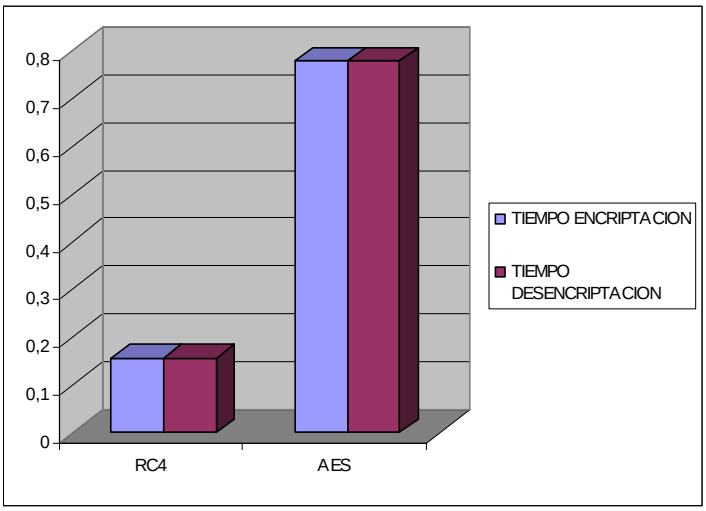
Encriptación: Algoritmo AES con 0,078 Bits

Desencriptación: Algoritmo AES con 0,094 Bits

Tamaño de Clave 128 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,156	0,156
AES	0,78	0,78

Tabla 4.6: Encriptación a 128 Bits



El resultado que se obtuvo con clave de 128 bits y los tiempos mínimos fue:

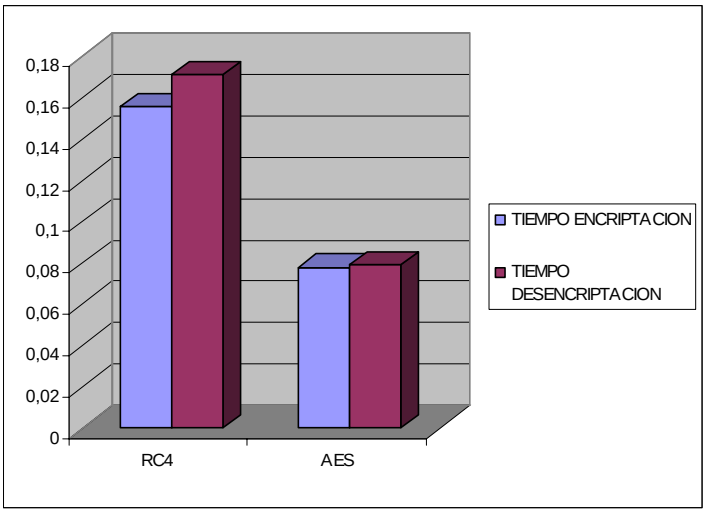
Encriptación: Algoritmo RC4 con 0,156 Bits

Desencriptación: Algoritmo RC4 con 0,156 Bits

Tamaño de Clave 256 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	0,156	0,171
AES	0,078	0,079

Tabla 4.7: Encriptación a 256 Bits



El resultado que se obtuvo con clave de 256 bits y los tiempos mínimos fue:

Encriptación: Algoritmo AES con 0,078 Bits

Desencriptación: Algoritmo AES con 0,079 Bits

4.5.3.- TEST 3

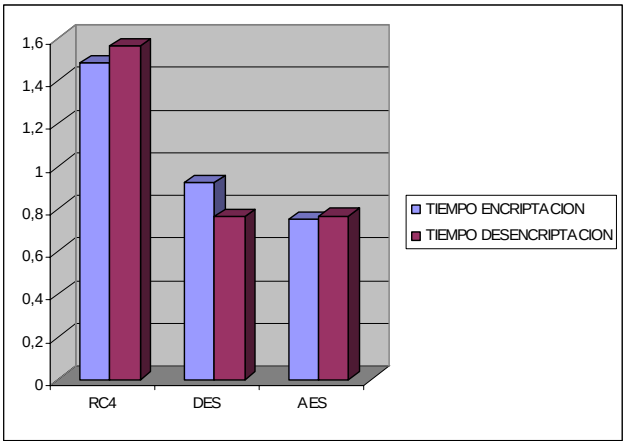
Tamaño de Archivo: **10 Mb**
Tamaño de Clave: **64 bits**

En este Test se trabajo con un archivo de tamaño 10 Mb, algoritmos RC4, DES y AES con clave de 64 bits, algoritmos RC4 y AES con claves de 128 y 256 bits, y sus tiempos de encriptación y desencriptación. Como se ve en las siguientes tablas y datos estadísticos

Tamaño de Clave 64 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	1,484	1,563
DES	0,922	0,765
AES	0,75	0,766

Tabla 4.8: Encriptación a 64 Bits



El resultado que se obtuvo con clave de 64 bits y los tiempos mínimos fue:

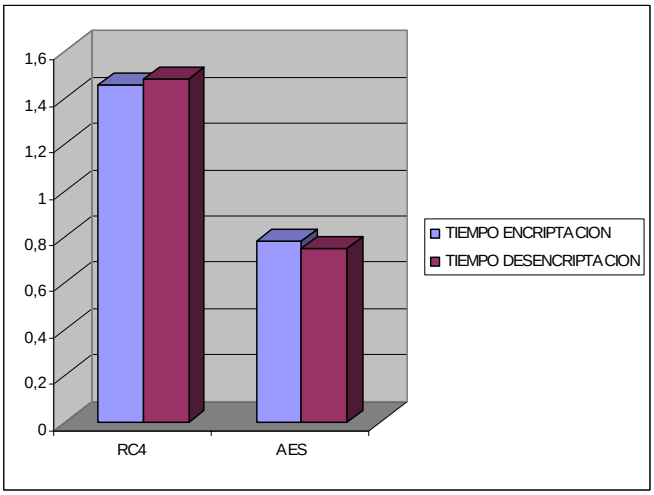
Encriptación: Algoritmo AES con 0,75 Bits

Desencriptación: Algoritmo DES con 0,765 Bits

Tamaño de Clave 128 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	1,453	1,484
AES	0,781	0,75

Tabla 4.9: Encriptación a 128 Bits



El resultado que se obtuvo con clave de 128 bits y los tiempos mínimos fue:

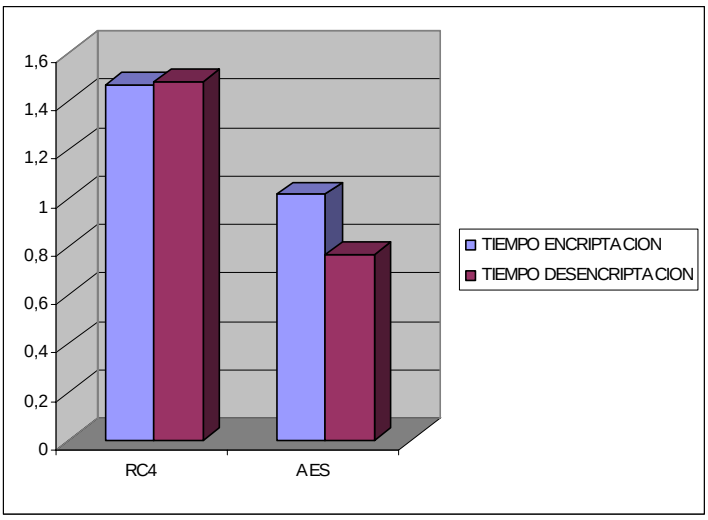
Encriptación: Algoritmo AES con 0,781 Bits

Desencriptación: Algoritmo AES con 0,75 Bits

Tamaño de Clave 256 bits

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION
RC4	1,469	1,484
AES	1,015	0,766

Tabla 4.10: Encriptación a 256 Bits



El resultado que se obtuvo con clave de 256 bits y los tiempos mínimos fue:

Encriptación: Algoritmo AES con 1,015 Bits

Desencriptación: Algoritmo AES con 0,766 Bits

4.6.- Análisis de resultados

Los resultados de la encriptación y desencriptación que se realizó con cada uno de los algoritmos como son RC4, DES y AES se evaluaron con respecto al tiempo y tamaño de los archivos.

Estos resultados finales lo veremos en las siguientes Tablas.

Archivo de 500 Kb y clave de 64 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
C4	-	-	-
DES	0,016	0,019	0,016
AES	-	-	-

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo DES, Tiempo de Encriptación 0,016

Archivo de 1 Mb y clave de 64 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
RC4	-	-	-
DES	-	-	-
AES	0,078	0,094	0,078

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo DES, Tiempo de Encriptación 0,078

Archivo de 10 Mb y clave de 64 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
RC4	-	-	-
DES	-	0,76	-
AES	0,75	-	0,75

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo AES, Tiempo de Encriptación 0,75

Archivo de 500 Kb y clave de 128, 256 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
RC4	-	-	-
AES	0,031	0,047	0,031

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo AES, Tiempo de Encriptación 0,031

Archivo de 1 Mb y clave de 128, 256 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
RC4	-	-	-
AES	0,078	0,079	0,078

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo AES, Tiempo de Encriptación 0,078

Archivo de 10 Mb y clave de 128, 256 bits:

	TIEMPO ENCRIPTACION	TIEMPO DESENCRIPTACION	Resultado
RC4	-	-	-
AES	0,75	0,78	0,75

Como resultado final en su tiempo mínimo obtenido es:

Resultado: Algoritmo AES, Tiempo de Encriptación 0,75

Conclusión:

Se ha concluido, con respecto a los resultados finales que el algoritmo **AES** encripta y desencripta en el menor tiempo y con el archivo de mayor tamaño, además de disponer de una clave de **256 bits**.