Laboratorio 8
Algoritmo DES: Paso a Paso

0. Introducción

El Data Encryption Standard (DES) es un algoritmo de cifrado de bloques de 64 bits con una clave de 56 bits, el cual usa funciones de Feistel en 16 ciclos. Aunque hoy en día DES no es considerado lo suficientemente fuerte para la mayoría de las aplicaciones tecnológicas, principalmente porque su tamaño de clave es muy pequeño, fue un estándar muy popular de cifrado en la industria hasta el 2002, cuando fue reemplazado por AES.
Se dice que las funciones de Feistel de DES y los ciclos aplicados tienen un "Efecto Avalancha" ("Avalanche Effect"), que es deseable en algoritmos de cifrado y funciones de hash.
Vamos a repasar la operación de DES paso a paso, y a analizar algunas de sus propiedades criptográficas.

Los objetivos de este laboratorio son:

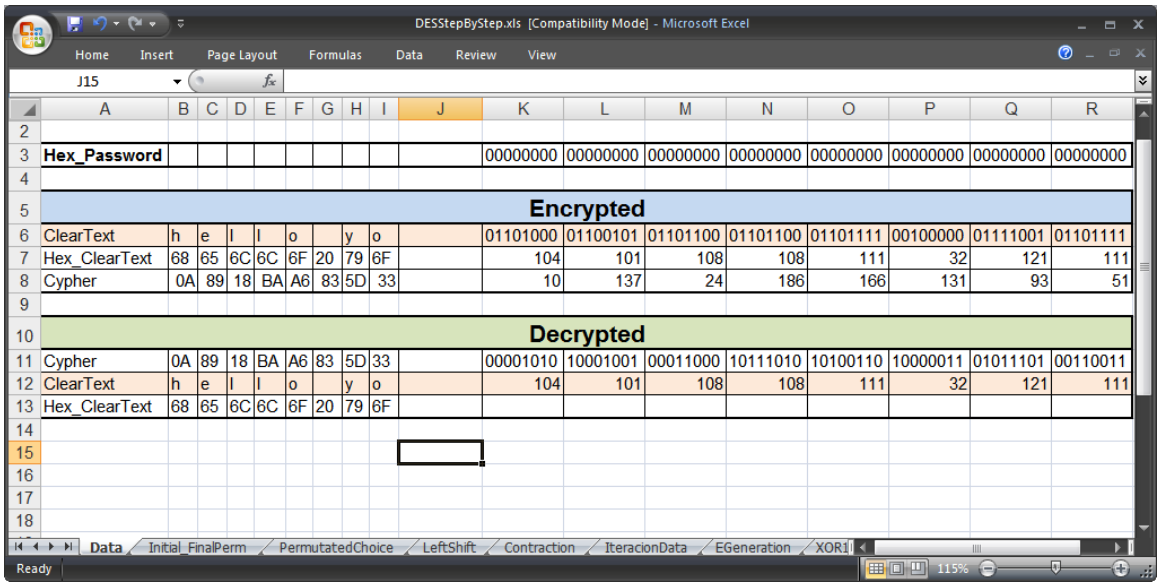
1. Analizar y entender algunas propiedades criptográficas de DES.
2. Repasar la operación de DES, paso a paso.

1. Requerimientos

Para este laboratorio usted necesita las siguientes herramientas:

1. MS Excel 2003 o superior. También sirve cualquier otra herramienta compatible (Open Office Spreadsheet, Star Office, etc.)
2. DESStepByStep.xls

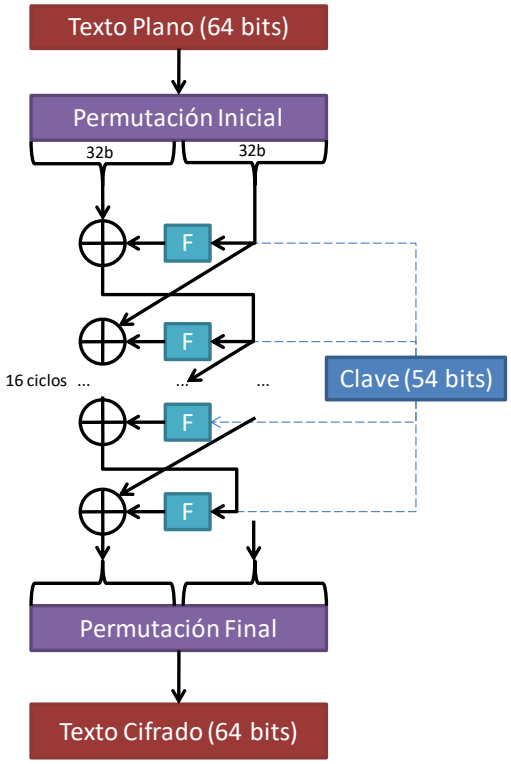
Abra DESStepByStep.xls en su herramienta de hoja cálculo. Debe ver algo como lo siguiente:



2. Repaso DES – Cifrado y Descifrado

Antes de entender el archivo DESStepByStep.xls vamos a hacer un resumen del funcionamiento del algoritmo DES. Recordemos DES es un algoritmo de bloques, cada bloque es de 64 bits (8 bytes), además se usa una clave de 56 bits (7 bytes) más un byte adicional que se suele usar para verificación.

La estructura de DES corresponde a una "Red de Feistel" (Feistel Network), que se puede entender mejor en la gráfica a continuación.



En esencia, lo que sucede es que entra el bloque de 64 bits, que corresponde al texto plano, al cual se le aplica una permutación inicial. Este bloque es procesado de acuerdo a una función que está parametrizada por la clave de 56 bits, por 16 ciclos (el símbolo ⊕ representa la operación XOR).
La función básicamente es una serie de sustituciones y permutaciones que van "revolviendo" los bits. Una vez ejecutados los 16 ciclos, se aplica una permutación final (que es exactamente inversa a la permutación inicial).

El resultado es otro bloque de 64 bits, totalmente diferente al inicial. Para descifrar el bloque, se aplica el mismo procedimiento en sentido contrario.

Tomemos DESStepByStep.xls y abrámoslo en la primera hoja. En la fila superior está la clave (en representación hexadecimal) con la que se ejecuta el procedimiento, a la derecha de ella está su correspondiente representación en binario.

Hex_Password	01	23	45	67	89	ab	cd	ef		00000001	00100011	01000101	0110

A continuación está el texto claro (en ASCII) que queremos cifrar (8 bytes), los cuales están debajo en su representación hexadecimal.

ClearText	N	o	w		i	s		t		01001110	01101111	01110111	00
Hex_ClearText	4E	6F	77	20	69	73	20	74		78	111	119	

En la siguiente fila al texto claro en hexadecimal, se encuentra el texto cifrado en hexadecimal.

Cypher	3F	A4	0E	8A	98	4D	48	15		63	164	14	

Introduzca diferentes valores de clave y de texto claro viendo cómo varía. Tenga en cuenta que la clave está en hexadecimal, mientras que el texto claro está en ASCII (es decir, en caracteres convencionales). Por ejemplo, el siguiente sería el resultado de cifrar el texto "cleartex" usando la clave "01 01 01 01 01 01 01 01":

Hex_Password	01	01	01	01	01	01	01	01
ClearText	c	l	e	a	r	t	e	x
Hex_ClearText	63	6C	65	61	72	74	65	78
Cypher	B1	22	9C	64	5D	8A	9B	E8

Pregunta 1: ¿Tendría sentido mostrar la representación ASCII del texto cifrado? ¿Por qué?

Ahora tome un texto cifrado cualquiera, y una clave cualquiera. Escoja un byte de la clave, y varíelo de manera que únicamente cambie el último bit de la representación binaria. Por ejemplo, supongamos que tenemos el texto "cleartex" y la clave "A0 A1 A2 A3 A4 A5 A6 A7" y escogemos el tercer byte (A2), para cambiar el último bit lo reemplazamos por A1, observemos las representaciones binarias:

Hex_Password	A0	A1	A2	A3	A4	A5	A6	A7		10100000	10100001	10100010	1

Hex_Password	A0	A1	A3	A3	A4	A5	A6	A7		10100000	10100001	10100011	10

Pregunta 2: ¿Cambia en algo el texto cifrado al hacer esta alteración? ¿Por qué?

Pregunta 3: ¿Tiene esto alguna relación con el hecho de que la clave de DES sea únicamente de 56 bits?

Intente haciendo diferentes cambios en cualquier bit de la clave, que

no sea el último de algún byte. Observe los cambios en el texto cifrado.

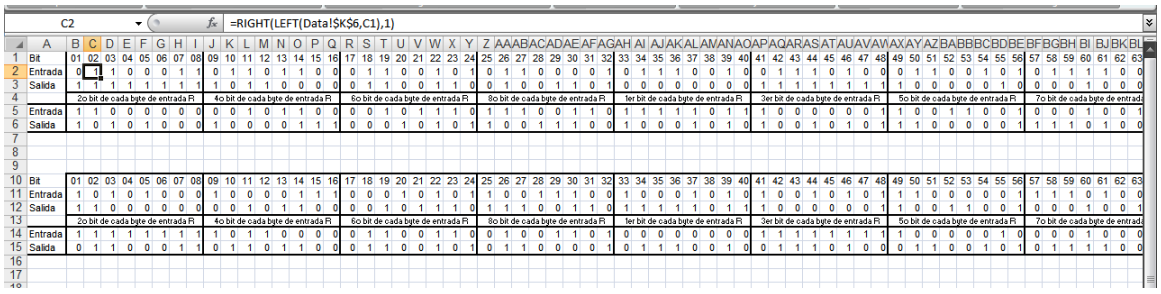
Pregunta 4: ¿Qué sucede? ¿Cuántos bytes del texto cifrado se alteran cuando se cambia un solo bit (que no es el último de algún byte)? Tome algunas capturas de pantalla (screenshots).

Ahora mantenga la clave constante, y varíe ligeramente el texto claro. Que primero cambie un byte cualquiera, y luego cambie un solo bit.

Pregunta 5: ¿Qué sucede? ¿Cuántos bytes del texto cifrado se alteran cuando se cambia un solo bit del texto plano? Tome algunas capturas de pantalla (screenshots).

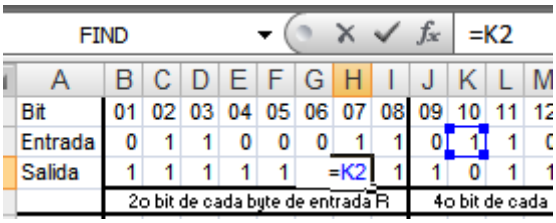
3. Permutaciones

Recordemos que DES ejecuta una permutación final, y una permutación inicial que fueron escogidas de manera arbitraria. Dichas permutaciones no agregan casi ninguna fortaleza criptográfica, al parecer fueron agregadas únicamente para facilitar la operación de datos en el hardware disponible en el momento de su creación (años 70's).
Para ver estas permutaciones en el archivo DESStepByStep.xls, vaya a la segunda hoja de cálculo (PermutaciónInicial_Final).



En la primera fila están los números de bit del texto claro (hasta 64), en la segunda fila están cada uno de los bits de ese texto. En la tercera fila están los mismos bits después de aplicar la permutación.

Observe que cada valor en la tercera fila se refiere a algún otro valor en la segunda fila:

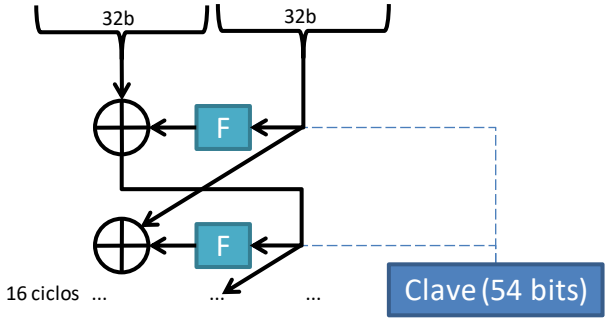


Pregunta 6: ¿Puede establecer un patrón en la permutación? ¿A qué corresponden los primeros 8 bits? ¿Los segundos? ¿Los demás?

Pregunta 7: ¿Por qué se dice que estas permutaciones no agregan ningún valor criptográfico? Explique.

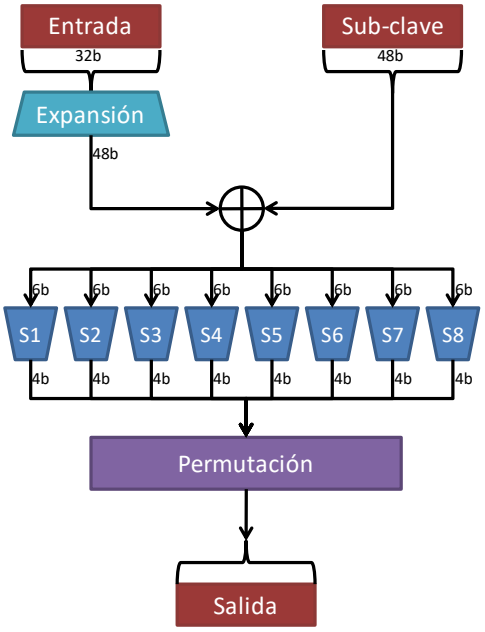
4. 16 Ciclos – Efecto Avalancha

Como ya vimos, después de hacer la permutación inicial, los datos son pasados a través de 16 ciclos, en el esquema llamado "Red de Feistel", en la que se aplica la operación XOR.



En la Red de Feistel hay una función (denotada como F en la figura) que opera sobre un bloque de 32 bits (la mitad de los datos). Dicha función es conocida como "Función de Feistel" y está parametrizada por la clave, a diferencia de las permutaciones inicial y final (que siempre son fijas).

Miremos la función de Feistel internamente:



Observe que se toma un bloque de 32 bits, y se le aplica una "Expansión" para generar un bloque de 48 bits. El resultado de esta expansión se le aplica la operación XOR con un fragmento de la clave (en cada ciclo es un fragmento diferente).
El resultado del XOR se parte en bloques de 6 bits, a cada uno de los cuales se les aplica una sustitución diferente (funciones S). Cada sustitución reemplaza 6 bits que entran por 4 bits de acuerdo a una tabla. Se dice que estas sustituciones son el corazón de la seguridad de DES.
Al resultado de las sustituciones se les aplica finalmente una permutación.

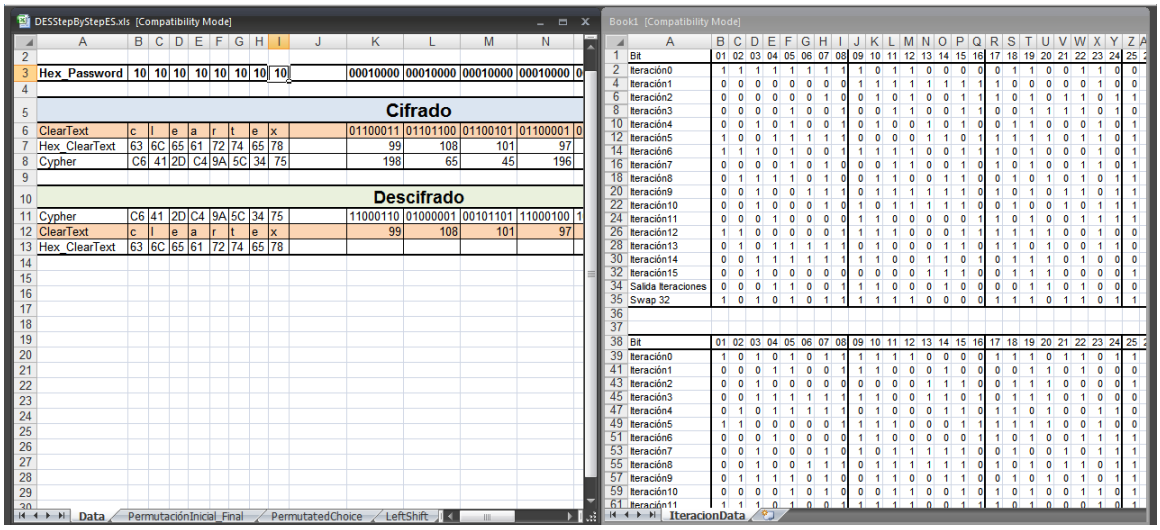
En el archivo DESStepByStep.xls, cada uno de los elementos de esta función está en una hoja de cálculo.

Pregunta 8: Trate de identificar en qué hoja de cálculo está cada uno de los siguientes elementos de la función F (tome screenshots):

1. ***La función de Expansión.***
2. ***El fragmento de la clave que se escoge en cada ciclo.***
3. ***El XOR entre el fragmento de la clave y el resultado de la expansión.***
4. ***Las funciones de sustitución.***
5. ***La permutación.***

Trate de abrir la primera hoja de cálculo (Data) y la 6ta (iteracionData) una al lado de la otra, como se muestra a

continuación:



En la hoja "iteracionData" se muestran los bloques de resultado de cada iteración.

Repita los experimentos que hizo en las preguntas 4 y 5, cambiando algunos bits en la clave, dejando el mensaje constante y viceversa. Copie los valores resultantes en una nueva hoja de cálculo, ejecute un cambio de un sólo bit, y copie los nuevos valores resultantes en la nueva hoja de cálculo.

Compare los resultados de antes y después de los cambios.

Pregunta 9: ¿Qué sucede con los datos de las iteraciones cuando cambia un bit? ¿Qué tan diferentes son las primeras iteraciones? ¿Qué tan diferentes son las últimas iteraciones? Muestre un par de screenshots con la comparación.

5. Claves Débiles

Existen un grupo de claves de DES que son conocidas como "claves débiles" (weak keys) y "claves semi-débiles" (semi-weak keys). Un ejemplo de clave débil en DES es "01 01 01 01 01 01 01 01". Veamos que sucede con esta clave.

Vaya a la primera hoja de cálculo (Data), introduzca algún texto claro, y la clave "01 01 01 01 01 01 01 01":

Hex_Password	01	01	01	01	01	01	01	01		00000001	00000001	00000001	00000001	00000001	00000001	00000001	00000001

Cifrado																	
ClearText	c	l	e	a	r	t	e	x		01100011	01101100	01100101	01100001	01110010	01110100	01100101	01111000
Hex_ClearText	63	6C	65	61	72	74	65	78		99	108	101	97	114	116	101	120
Cypher	C6	41	2D	C4	9A	5C	34	75		198	65	45	196	154	92	52	117

Descifrado																	
Cypher	C6	41	2D	C4	9A	5C	34	75		11000110	01000001	00101101	11000100	10011010	01011100	00110100	01110101
ClearText	c	l	e	a	r	t	e	x		99	108	101	97	114	116	101	120
Hex_ClearText	63	6C	65	61	72	74	65	78									

A continuación vaya a la hoja de cálculo 4 (leftShift), y observe los datos. Si respondió correctamente la pregunta 8, entonces debe poder interpretar lo que sucede en esta hoja.

Pregunta 10: Explique qué sucede con las sub-claves de cada iteración. ¿Son iguales? ¿Son diferentes?

Otros ejemplos de claves débiles son:
"E0 E0 E0 E0 F1 F1 F1 F1"
"1F 1F 1F 1F 0E 0E 0E 0E"

Pregunta 11: ¿Podría explicar en qué consiste una "clave débil"? ¿Qué otras claves débiles se le ocurren?

Las claves semi-débiles, por el contrario, son parejas de claves. Estas parejas de claves también tienen propiedades interesantes en cuanto a las sub-claves generadas para cada iteración. Algunas parejas de claves "semi-débiles" son:

"01 1F 01 1F 01 0E 01 0E" y "1F 01 1F 01 0E 01 0E 01"
"01 E0 01 E0 01 F1 01 F1" y "E0 01 E0 01 F1 01 F1 01"
"01 FE 01 FE 01 FE 01 FE" y "FE 01 FE 01 FE 01 FE 01"

Utilice estas claves para cifrar un texto, y observe qué sucede en la hoja 4 (leftShift).

Pregunta 12: ¿Qué sucede con las sub-claves de cada iteración al usar claves semi-débiles?

6. Conclusiones

DES es un algoritmo criptográfico "simétrico" o de clave "secreta", que fue utilizado como estándar durante mucho tiempo, sin embargo ya cayó en desuso porque su tamaño de clave es muy pequeño, lo que lo

hace vulnerable a un ataque de "fuerza bruta".

Pregunta 13: Suponga que un procesador como el de su PC necesita 20 ciclos de procesador para cifrar un bloque de 64 bits con DES. (16 ciclos para cada iteración, 2 ciclos para la permutación inicial, y 2 ciclos para la permutación final).
¿Cuánto demoraría un ataque de fuerza bruta, con un tamaño de clave de 56 bits? Muestre los cálculos usando la velocidad del procesador de su PC.

Observe de nuevo la Red de Feistel que compone DES. Suponga que IZ_i representa el bloque izquierdo resultante de la i-ésima iteración, mientras que DR_i representa el bloque derecho. En cada iteración se cumple que:

$IZ_i = DR_{i-1} \oplus F_i(IZ_{i-1})$
$DR_i = IZ_{i-1}$

Recuerde que $\oplus$ representa la operación XOR, y que F_i es diferente en cada iteración porque se usa un fragmento diferente de la clave.
Para descifrar un mensaje DES, basta con aplicar la misma Red de Feistel, únicamente invirtiendo el orden de los fragmentos de la clave.

Pregunta 14: Intente explicar por qué este procedimiento funciona para descifrar un archivo. Tenga en cuenta la siguiente identidad:

$$A \oplus A \oplus B = B$$

Pregunta 15: Teniendo en cuenta la pregunta anterior. ¿Qué efectos tiene una clave débil sobre las funciones F_i en cada iteración? ¿Qué pasaría si cifráramos un mensaje con una clave débil, y el resultado lo volviéramos a cifrar con la misma clave débil?

Pregunta 16: ¿Qué efectos tiene una clave semi-débil sobre las funciones F_i en cada iteración? ¿Qué pasaría si cifráramos un mensaje con una clave semi-débil, y el resultado lo cifráramos con su pareja?