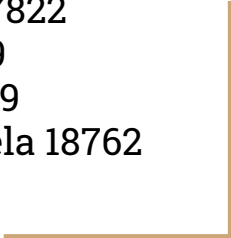




Laboratorio 2

Block Ciphers

Juan Fernando De León 17822
Jose Amado Garcia 181469
Edgar Andree Toledo 18439
Ricardo Antonio Valenzuela 18762
Sara Zavala 18893



Parte 1

Investigación

¿Por qué no usamos 2DES?

¿Cual es el problema con este tipo de cifrado?

La seguridad de DES depende del tamaño de la clave. DES tiene una clave de 56 bit, sin embargo, al aplicar el algoritmo 2 veces únicamente se logra duplicar el tiempo para romper 2DES usando la “fuerza bruta”. 2DES tiene una seguridad de 2^{57} .

Meet-in-the-middle Attack

Como se mencionó anteriormente, 2DES únicamente encripta 2 veces. Después de descifrar con cada clave, se verifica si hay coincidencias con las salidas almacenadas de los posibles cifrados. Cuando tenemos una coincidencia, hemos localizado un par de claves posiblemente correcto. Generalmente, el número de tuplas encontradas es reducido, por lo que probar una por una no es un proceso exhaustivo.

Implementación DES

- Inicio
- Crea 16 subclaves, cada una de 48 bits
 - Realizar las permutaciones de cada clave en base a la tabla PC-1
 - Por cada clave, dividir la cadena de 56 bits en dos (C_n , D_n).
 - Dependiendo de la ronda, realizar n corrimiento de bits (XOR)
 - Unimos nuevamente la cadena para obtener la clave K_n . ($K_n = C_n \oplus D_n$).
 - Realizar las permutaciones de cada clave en base a la tabla PC-2
- Cifrar mensaje por cada bloque de 64 bits
 - Aplicar la permutación a la cadena de 64 en base a la tabla IP.
 - Se divide la cadena de 64 bits en dos (L , R).
 - Tomamos R_n y se vuelve L_{n+1} .
 - Tomamos L_n y realizamos XOR con $f(R_n, K_{n+1})$.

- $f(R_n, K_{n+1})$
 - Se toma R_n y se expande a 48 bits aplicando la permutación de la tabla E.
 - $K_{n+1} \text{ XOR } E(R_n) = B1B2B3B4B5B6B7B8$
 - Se aplica las "S boxes" a cada B_n . ($S_n(B_n)$)
 - A
 $S1(B1)S2(B2)S3(B3)S4(B4)S5(B5)S6(B6)S7(B7)S8(B8)$ aplicamos la permutación en base a la tabla P.
 - Retornamos el resultado
- Al finalizar la ronda 16 tenemos L_{16} y R_{16} .
- Invertimos el orden R_{16} , L_{16}
 - Aplicamos la permutación inversa de la tabla IP (IP^{-1}).
- Retornamos el resultado del paso anterior, mensaje encriptado en hexadecimal.

Implementación AES

- La llave se expande haciendo uso del esquema de claves de Rijndael (AES key schedule), se obtiene una llave de 128 bits.
- Primera ronda:
 - Los 16 bytes del texto se convierten en 128 bits, a estos se les hace XOR con la llave generada. (AddRoundKey)
- Rondas:
 - Ocurren sustituciones de bits, donde un bit se mueve de posición acorde a una tabla, también llamada S-Box. El resultado es una matriz de 4x4. (SubBytes)
 - Cada fila de la matriz es corrida hacia la izquierda, los pedazos que se “desborden” son reingresados por la derecha. (ShiftRows)
 - La primera fila no se corre.
 - La segunda fila se corre 1 bit.
 - La tercera se corre 2 bits.
 - La cuarta se corre 3 bits

- Cada columna de la matriz, que está conformada por 4 bytes, es convertido a un nuevo set de bytes haciendo uso de una transformación lineal. (MixColumns)
 - Se realiza de nuevo un AddRoundKey
- Última ronda:
 - En esta ronda se realizan todos los pasos anteriores menos el MixColumns. Es decir se realiza:
 - SubBytes
 - ShiftRows
 - AddRoundKey

SubBytes

Como su nombre lo indica es una sustitución de bytes, básicamente los 16 bytes de entrada son sustituidos haciendo uso de una tabla llamada S-box. AES a diferencia del DES utiliza la misma tabla para todos sus bytes.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	1	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
40	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	2	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	6	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	8
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	3	16	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

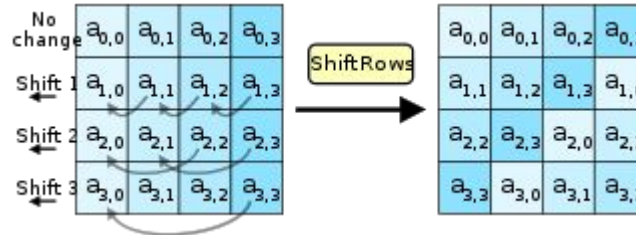
S-box, extraída de:

<https://www.commonlounge.com/discussion/e32fdd267aaa4240a4464723bc74d0a5>

ShiftRows

Es un paso simple donde las filas de la matriz de bytes se van corriendo hacia la izquierda dependiendo de qué fila sea.

- La primera fila no se corre.
- La segunda fila se corre 1 bit.
- La tercera se corre 2 bits.
- La cuarta se corre 3 bits



Funcionamiento de ShiftRows, extraída de:

<https://upload.wikimedia.org/wikipedia/commons/thumb/6/66/AES-ShiftRows.svg/320px-AES-ShiftRows.svg.png>

MixColumns

Esta función recibe 4 bytes de una columna de la matriz y devuelve otros 4 bytes. Durante este paso se toman los 4 bytes de input para multiplicarlos por una matriz y obtener el nuevo set de 4 bytes. Este paso no se realiza en la última ronda.

$$\begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \end{bmatrix} = \begin{bmatrix} 14 & 11 & 13 & 9 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Ejemplo de la multiplicación de bytes con la matriz:

<https://asecuritysite.com/aes04.png>

Referencias

- <https://www.nku.edu/~christensen/3DES.pdf>
- <http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm>
- https://www.tutorialspoint.com/cryptography/advanced_encryption_standard.htm
- <https://www.commonlounge.com/discussion/e32fdd267aaa4240a4464723bc74d0a5>
-