UNIVERSIDAD DE COSTA RICA
Escuela de Ingeniería Eléctrica
IE0217 – Estructuras abstractas de datos y algoritmos para ingeinería

Proyecto 1: Comparación de los algoritmos de cifrado RSA y DES
Grupo 04

Fabián Guerra Esquivel, B53207
Yeison Rodríguez Pacheco, B56074

24/11/18

Índice

1. Introducción

La encriptación es el proceso para codificar información, de tal manera que, idealmente, solo las partes con acceso permitido pueden acceder a tal información. Para encriptar información, se utlizan algoritmos de cifrado, los cuales funcionan convirtiendo la información por encriptar en un cifrado, el cual solamente puede ser leído si es desencriptado, utilizando generalmente una llave.

1.1. Claves simétricas y no simétricas

El tipo de cifrado más sencillo es el simétrico. En éste, se requiere únicamente una sola clave tanto para encriptar como para desencriptar. El algoritmo DES, que se desarrollará en este proyecto es de tipo simétrico. Por otro lado, están algoritmos de cifrado más complejos, en los cuales la clave para encriptar la información es diferente, pero relacionada, de la utilizada para desencriptarla. Este método de encriptación se conoce también como cifrado de clave pública, debido a que, si no existe manera de obtener una clave teniendo la otra, la clave pública se puede ser conocida por cualquier usuario, sin embargo, no se puede desencriptar el mensaje sin el uso de la clave privada. El algoritmo RSA, por desarrollar en este proyecto, utiliza claves no simétricas.

2. Reseña del algoritmo/estructura

2.1. DES

El algoritmo DES, (Data Encryption Standard) es de los algoritmos de encriptación simétrica más utilizados a nivel mundial. DES funciona convirtiendo en bloques de 64 bits de código hexadecimal de texto y rellena con ceros el resto, en caso de no ser múltiplo de 64. El output de cada bloque también es de 64 bits, y las llaves utilizadas son de 56 bits efectivos, que se guardan en 64 bits (cada octavo bit es irrelevante en la llave). DES tiene el siguiente algoritmo: [1] [2]

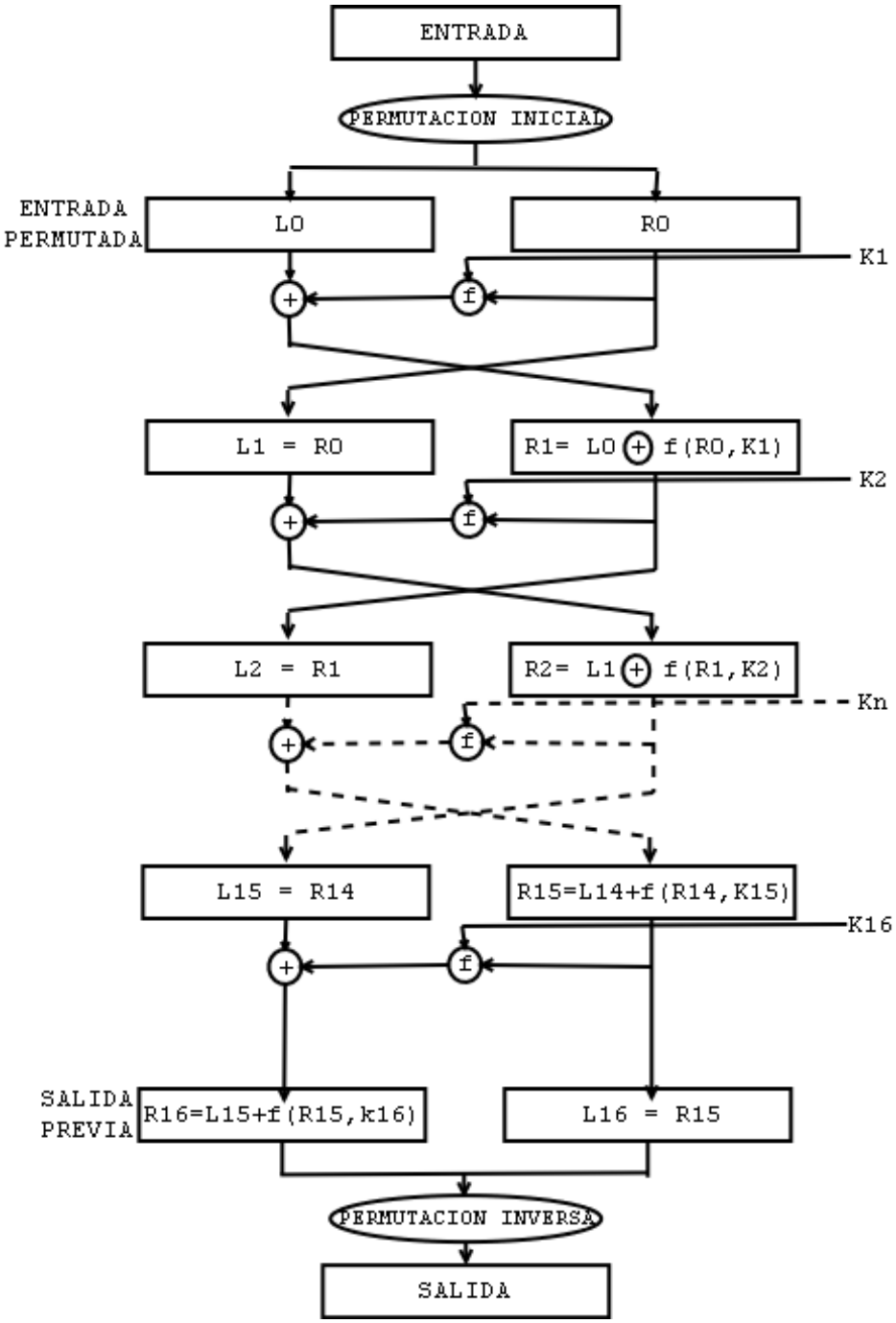

Figura 1: Diagrama de flujo del algoritmo DES [1]

Paso 1: generar 16 sub-claves, de 48 bits cada una:

- De la llave original de 64 bits, se descartan los bits 8, 16, 24, 32, 40, 48, 56 y 64, para obtener una llave de 56 bits
- Se permuta la llave de acuerdo con la figura 2
- Se separa la llave de 56 bits en dos mitades de 28 bits: izquierda y derecha

- Cada mitad hace un shift circular de sus bits para cada una de las 16 rondas; las rondas 1, 2, 9 y 16 hace un shift a la izquierda sencillo, y el resto de las rondas hace un shift doble.
- Se unen las dos mitades nuevamente para tener la llave de 56 bits, y se permutan de acuerdo a la tabla de la figura 3 para tener una llave de 48 bits
- Se repite este proceso 16 veces para obtener las 16 llaves K_i que se muestran en el algoritmo de la figura 1

Parity-bit drop table

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Figura 2: Tabla de reducción de la clave de 64 bits a 56 bits

Key-compression table

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figura 3: Tabla de compresión de la llave de 56 bits a 48 bits

Paso 2: Permutación inicial y permutación final Cada bloque de 64 bits de la entrada por codificar pasa por dos permutaciones, las cuales son inversas una de la otra, es decir, si en la permutación inicial, el i-ésimo bit de entrada corresponde al j-ésimo bit de salida, entonces en la permutación final, el j-ésimo bit de entrada corresponde al i-ésimo bit de salida. Este paso no tiene ningún valor criptográfico.

Paso 3: 16 rondas idénticas Como se puede observar en la figura 1, luego de tener la entrada permutada, se hacen las rondas, las cuales siguen los siguientes pasos:

- Se toman los bloques izquierdo L_{i-1} y derecho R_{i-1} de la ronda anterior.
- La parte derecha R_{i-1} pasa a ser la parte izquierda L_1

- La parte derecha R_{i-1} junto con la llave K_i son argumentos de la función DES, cuya salida se hace XOR con L_{i-1} para crear el nuevo bloque R_i

La función DES: Esta función opera sobre el bloque derecho de 32 bits de la ronda anterior y la llave de la ronda actual, de 48 bits, para generar una salida de 32 bits. Hace los siguientes pasos:

- Se expanden los 32 bits del bloque a 48 bits, utilizando la tabla de expansión de la figura 4
- Se hace XOR del resultado de la expansión con la llave de la ronda
- Se dividen los 48 bits en 8 grupos de 6, los cuales son ingresados cada uno en un S-box, el cual hace una mezcla y genera un output de 4 bits. La tabla que guía esta operación se encuentra en los anexos.

Expansion D-box table

32	01	02	03	04	05
04	05	06	07	08	09
08	09	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	31	31	32	01

Figura 4: Tabla de expansión de 32 bits a 48 bits

Al terminar las 16 rondas, se juntan los dos bloques de 32 bits y se hace la permutación final, con lo que se tienen 64 bits. Estos se juntan con el resto del texto encriptado.

2.2. RSA

El algoritmo de encriptación RSA, a diferencia del algoritmo DSA es asimétrico, por lo que las claves de encriptación y decriptación son distintas, y se cumple la condición de que es imposible obtener la llave privada a partir de la llave pública. El algoritmo se basa principalmente en la dificultad para factorizar números que son producto de números primos muy grandes.
El algoritmo funciona de la siguiente manera: [3] [4]
Paso 1: Generación de las llaves

- Se deben elegir o generar dos números primos p y q, preferiblemente muy grandes y con poca diferencia entre ellos, tal que su multiplicación $n = p * q$ quepa en el espacio designado para la llave. Los valores típicos de llaves para RSA son de 1024 bits o 2048 bits.
- Obtener $n = p * q$ y $\Phi = (p-1) * (q-1)$

- Elegir un número entero e tal que:
  $1 < e < \Phi$
  $mcd(e, \Phi) = 1$
- Se calcula el exponente privado, para algún k:
  $d = (k * \Phi + 1)/e$
- De esta manera, se obtienen ambas claves:
  - **Clave pública:** (n, e)
  - **Clave privada:** (p, q, d)

Paso 2: Encriptación de los datos
El algoritmo RSA únicamente opera sobre números enteros, por lo que, en caso de querer encriptar texto, se debe utilizar el código ascii de cada letra. Si se quiere encriptar un número x, el dato encriptado c es:

$$c = x^e \mod n \tag{1}$$

Paso 3: Desencriptación de los datos
Para obtener el dato desencriptado x, a partir del dato encriptado c se tiene la ecuación:

$$x = c^d \mod n \tag{2}$$

3. Experimentos y pruebas que se realizarán

3.1. Comparación de ambos algoritmos en complejidad computacional

Para realizar la comparación de los tiempos de computación de ambos algoritmos, se harán muchas iteraciones de ambos programas, modificando ciertos valores para observar las diferencias. En el caso de ambos algoritmos, se harán pruebas modificando la longitud del texto por encriptar, haciendo varias repeticiones del ciclo, para evitar aberraciones en los tiempos de ejecución. Posteriormente, se realizarán ajustes a la curva obtenida, para obtener la complejidad del algoritmo en relación con la longitud del texto.

Por otro lado, se pueden hacer pruebas individuales para analizar la complejidad del algoritmo RSA, con respecto a la longitud de la clave utilizada. Como el DES tiene llaves con longitud determinada, no se puede realizar esta prueba. Caso contrario es la RSA, pues se pueden elegir muchos tamaños de claves, generalmente 1024 bits, o múltiplos de éste, pero pueden ser mucho menores.

3.2. Efectividad de la encriptación

Debido a su naturaleza, el cifrado DES es casi imposible de quebrar utilizando una computadora de uso general como la utilizada para realizar pruebas de su efectividad, sin embargo, las claves RSA, al tener una longitud de clave variable, se puede probar quebrarlas, utilizando llaves y texto cortos.

Referencias

[1] *Data Encryption Standard*. Cleveland State University. Book chapter 6. [En línea] Recuperado de: https://academic.csuohio.edu/yuc/security/Chapter_06_Data_Encription_Standard.pdf

[2] Orlin, J. (2006) *The DES Algorithm Illustrated*. [En línea] Recuperado de: http://page.math.tu-berlin.de/ kant/teaching/hess/krypto-ws2006/des.html

[3] *RSA Algorithm*. [En línea] Recuperado de: https://www.di-mgt.com.au/rsa_alg.html#note3

[4] *Algorithm in Cryptography*. [En línea] Recuperado de: https://www.geeksforgeeks.org/rsa-algorithm-cryptography/

Anexos

Table 6.3 *S-box 1*

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Table 6.4 *S-box 2*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	01	08	14	06	11	03	04	09	07	02	13	12	00	05	10
1	03	13	04	07	15	02	08	14	12	00	01	10	06	09	11	05
2	00	14	07	11	10	04	13	01	05	08	12	06	09	03	02	15
3	13	08	10	01	03	15	04	02	11	06	07	12	00	05	14	09

Table 6.5 *S-box 3*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	00	09	14	06	03	15	05	01	13	12	07	11	04	02	08
1	13	07	00	09	03	04	06	10	02	08	05	14	12	11	15	01
2	13	06	04	09	08	15	03	00	11	01	02	12	05	10	14	07
3	01	10	13	00	06	09	08	07	04	15	14	03	11	05	02	12

Table 6.6 *S-box 4*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	07	13	14	03	00	6	09	10	1	02	08	05	11	12	04	15
1	13	08	11	05	06	15	00	03	04	07	02	12	01	10	14	09
2	10	06	09	00	12	11	07	13	15	01	03	14	05	02	08	04
3	03	15	00	06	10	01	13	08	09	04	05	11	12	07	02	14

Table 6.7 *S-box 5*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	02	12	04	01	07	10	11	06	08	05	03	15	13	00	14	09
1	14	11	02	12	04	07	13	01	05	00	15	10	03	09	08	06
2	04	02	01	11	10	13	07	08	15	09	12	05	06	03	00	14
3	11	08	12	07	01	14	02	13	06	15	00	09	10	04	05	03

Table 6.8 *S-box 6*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	01	10	15	09	02	06	08	00	13	03	04	14	07	05	11
1	10	15	04	02	07	12	09	05	06	01	13	14	00	11	03	08
2	09	14	15	05	02	08	12	03	07	00	04	10	01	13	11	06
3	04	03	02	12	09	05	15	10	11	14	01	07	10	00	08	13

Table 6.9 *S-box 7*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	00	08	13	03	12	09	07	05	10	06	01
1	13	00	11	07	04	09	01	10	14	03	05	12	02	15	08	06
2	01	04	11	13	12	03	07	14	10	15	06	08	00	05	09	02
3	06	11	13	08	01	04	10	07	09	05	00	15	14	02	03	12

Table 6.10 *S-box 8*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	10	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	10	15	03	05	08
3	02	01	14	07	04	10	8	13	15	12	09	09	03	05	06	11