UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA, 2023-1

CRIPTOGRAFÍA

ACTIVIDAD DE INVESTIGACIÓN

Simplified DES (sDES)

PROFESORA:

Dra. Rocío Aldeco Pérez

ALUMNA:

Karla Andrea Najera Noyola

Simplified DES (sDES)

Instrucciones

a. Utilizando la descripción y el pseudocódigo provistos, discuta cómo se debe implementar este algoritmo, cuál sería el mejor lenguaje de programación para hacerlo.
b. Discuta cuál es el proceso que debe seguir para descifrar los datos. Descríbelo y crea el pseudocódigo.
c. Subir un archivo con todos estos hallazgos (individual).
d. Realice un envío individual de su implementación utilizando el lenguaje de programación de su elección. Los casos de prueba son los presentados en la siguiente tabla:

Key	Plaintext	Ciphertext
0000011111	01010101	11000100
0000000000	00000000	11110000
1111111111	11111111	00001111

Introducción

Con el paso de los años la criptografía ha ido evolucionado con el fin de brindar soluciones cada vez más robustas, dando lugar a lo que hoy se conoce como la Criptografía Moderna. Esta rama fue creada en1948 a raíz de la Segunda Guerra Mundial y tomando como referencia la Teoría de la Información de Claude Shannon cuyo postulado principal era que ningún texto cifrado debía revelar ningún tipo de información acerca del texto sin formato, lo que implica que ningún atacante debería conocer información relacionada que sirva para su descifrado.

Hoy en día se ha vuelto crucial el uso de algoritmos que permitan la transferencia segura de datos a través de canales que no siempre pueden tener las máximas medidas para garantizar que la información cumpla con la triada de la seguridad. Por ello, es cada vez más frecuente la aparición de nuevos algoritmos con un mayor nivel de complejidad. Tomando en cuenta lo anterior, dentro de la Criptografía Moderna existirá la siguiente clasificación:

- **Algoritmos asimétricos**: Son aquellos métodos que se basan en una codificación de información haciendo uso de 2 claves: una privada y una pública, con el fin de que el remitente conserve la clave privada y cualquier receptor pueda recibir la clave pública. Se les considera relativamente nuevos.
- **Algoritmos simétricos:** Son métodos más simples que los asimétricos debido a que utilizan la misma llave para las funciones de encriptación y desencriptación. Esta clave debe ser compartida con todas las personas que requieren recibir el mensaje.

En esta práctica nos enfocaremos en el funcionamiento de DES, uno de los algoritmos simétricos más reconocidos en la actualidad, revisando su funcionamiento general y características importantes, además de generar su implementación en un lenguaje de programación.

Marco Teórico

La presente práctica hará uso de los siguientes conceptos:

- **DES**

DES (Data Encryption Standard) es un algoritmo de cifrado simétrico que fue escogido como un estándar FIPS en Estados Unidos en 1976 y creado por el Departamento de Comercio y la Oficina Nacional de Estándares en colaboración con la empresa IBM. Fue creado con el objetico de proporcionar al público general un algoritmo normalizado para redes de computadoras ya que fue basado en la aplicación de todos los conceptos criptográficos importantes del momento.

Su uso se fundamenta en un sistema monoalfabético, con un algoritmo de cifrado que aplica sucesivamente diversas permutaciones y sustituciones. En el inicio, el texto plano cifrar se somete a una permutación, con bloque de entrada de 64 bits (o múltiplo de 64), para posteriormente ser sometido a dos funciones principales, una función de permutación con entrada de 8 bits y otra de sustitución con entrada de 5 bits, en un proceso que consta de 16 etapas de cifrado. Cabe destacar que DES utiliza una clave simétrica de 64 bits de los cuales, 56 son usados para la encriptación. Mientras que los 8 que restan son de paridad para detectar errores en el proceso.

Tomando en cuenta el tamaño de esta clave, son posible un total de 2^{56} = 72.057.594.037.927.936 claves posibles, es decir, unos 72.000 billones de claves, por lo que la ruptura del sistema por fuerza bruta o diccionario es sumamente improbable, aunque no imposible si se dispone de suerte y una gran potencia de cálculo.

Hoy en día se le considera un algoritmo inseguro dado que su tamaño de clave es considerado corto e incluso muchas de estas han sido posibles de romperse en menos de 24 horas. Pese a que se dice que su variante TripleDES es más resistente en la práctica, el algoritmo fue sustituido eventualmente por AES (Advanced Encryption Standard).

- **S-Box**

Es uno de los componentes fundamentales en los algoritmos de cifrado de clave simétrica y en el caso de los cifradores por bloque (como es el caso de DES o AES) se utilizan para proteger y realizar la correspondencia entre un texto plano y uno cifrado. Suelen ser elegidas de tal forma que resistan el criptoanálisis y garanticen la seguridad de los mensajes.

Por lo general, las S-Box cuentan con una entrada de m bits y una salida de n bits haciendo uso de técnicas de sustitución

- **Funciones XOR**

Una función XOR se basa en el uso de la compuerta XOR u OR exclusiva, la cual realiza una función booleana A'B+AB'. Es decir, se basa en el funcionamiento de la siguiente tabla:

Entrada A	Entrada B	Salida S
0	0	0
0	1	1
1	0	1
1	1	0

Cabe destacar que uno de sus usos comunes es la implementación de la adición binaria, además de como un comparador o inverso condicional. Asimismo, en el terreno de la criptografía permite la generación de números pseudoaleatorios, teniendo como ejemplo los registros de desplazamiento de realimentación lineal.

El símbolo por el cual se identifica esta función es un signo de más encerrado dentro de un círculo.

Hipotesis

Dado que para esta práctica trabajaremos sobre una versión simplificada de DES que hace uso de una cadena de 8 bits y una clave de 10 bits, el proceso será menos largo y complejo. No obstante, retoma del algoritmo principal las siguientes 2 funciones fundamentales:

- ***Obtención de subkeys***

Esta función es la encargada de obtener 2 claves de 8 bits a partir de una clave original de 10 bits. Dado que es un cifrado de clave simétrica se utiliza la misma clave para cifrar y descifrar, por lo que ambas partes (emisor y receptor) deben poseerla.

El algoritmo que ejemplifica la obtención de las claves para esta versión simplificada y educativa de DES es el siguiente:

1.- Permutación de la clave de 10 bits.
2.- División en 2 subcadenas.
3.- Desplazamiento de un bit hacia la izquierda.
4.- Unión de las cadenas desplazadas
5.- Obtención de la subkey 1 mediante permutación a 8 bits.
6.-Desplazamiento de 2 bits del resultado del paso 3
7.- Unión de las cadenas obtenidas tras desplazamiento.
8.- Obtención de la subkey2 mediante permutación de 8 bits.

- **Función Feistel**

Esta función es un método de cifrado en bloque, creado por Horst Feistel, el cual es utilizado por un gran número de algoritmos de bloque. Entre sus características se encuentra que es reversible, por lo que funciona de la misma forma para cifrado y descifrado, requiriendo solamente invertir el orden de las subclaves que se utilicen.

En el caso de esta versión de DES simplificado, los pasos generales que se utilizarán para cifrar/descifrar una cadena son los siguientes:

1.- Dividir la cadena en 2 partes: left y right.
2.- Mezclar right con subkey1. Para ello, expandir right y aplicar XOR con la subkey correspondiente.
4.- Realizar una división temporal de la cadena.
5.- Aplicar función de mezcla con una S-Box S0.
6.- Aplicar función de mezcla con una S-Box S1.
7.- Concatenar resultados de los 2 pasos anteriores.
8.- Permutación del resultado.
9.- Aplicación de XOR a left y al resultado permutado.
10.- Concatenación de resultados.

Una vez conocido lo anterior, una primera hipótesis del proceso de cifrado podría mostrar que DES utiliza las funciones anteriores en conjunto con otra serie de intercambios y permutaciones. En el caso del proceso de descifrado se seguirán los mismos pasos, pero variando la subclave en el proceso de Feistel. Por ello, es realmente importante contar con la clave correcta en los 2 procesos debido a que algún dígito incorrecto o alterado no generará la cadena de salida de manera adecuada (ya sea cifrada o como texto plano).

Tomando en cuenta todo lo anterior, a continuación, profundizaremos en las características de DES y su implementación en un lenguaje de programación.

Desarrollo

Propuesta del proceso de cifrado

En primer lugar, se inicia el cifrado con la obtención de las subclaves a partir de la clave brindada por el usuario. Tras ello. Se realiza una permutación inicial del texto plano que le permite a la cadena estar lista para la aplicación de la función Feistel con la subkey 1. Acto seguido, se realiza un intercambio de posiciones de la cadena obtenida en el paso anterior y se procede a la segunda aplicación de fesitel, pero ahora con la subkey 2. Se realiza una permutación final del texto obtenido y se obtiene el mensaje cifrado. Los anteriores pasos se ejemplifican de la siguiente forma:

1.- Obtención de subkeys.
2.- Permutación inicial.
3.- Aplicación de Feistel en subkey 1.
4.- Intercambio de valores de la cadena.
5.- Aplicación de Feistel en subkey 2.
6.- Permutación final.

Proceso de descifrado a partir del cifrado y del uso de la clave

Para el proceso de descifrado, se realizan básicamente los mismos pasos que para la encriptación, pero intercambiando el orden de las funciones Feistel (en primer lugar se usa la subkey 2 y luego la subkey 1), por lo que el proceso general queda de la siguiente forma:

1.- Obtención de subkeys.
2.- Permutación inicial.
3.- Aplicación de Feistel en subkey 2.
4.- Intercambio de valores de la cadena.
5.- Aplicación de Feistel en subkey 1.
6.- Permutación final.

Implementación del algoritmo en el lenguaje de programación Java

Se selecciono Java para la implementación de este algoritmo debido a la capacidad de uso que presentan las clases envolventes, además del manejo de arreglos numéricos. Por ende, se usará el paradigma orientado a objetos.

Se hará uso de 2 clases por medio de las cuales un usuario podrá cifrar y/o descifrar sus mensajes:

- **Main.java:** Es la clase principal del proyecto. A través de esta se harán las interacciones con el usuario, incluyendo la captura de la clave y un mensaje, así como el cifrado/descifrado tras la aplicación del algoritmo. Se compone de un único método (Main) en el cual se imprime el menú de usuario y se realiza el flujo del programa.
- **DES.java:** Dentro de esta clase se definen todos los procesos y funciones que permiten el cifrado y descifrado de los mensajes que el usuario ingrese haciendo uso de los conceptos de DES. Se compone de las siguientes 6 variables de instancia:
    - Clave[]: Arreglo de enteros que posee la clave con la que el texto será cifrado.
    - Texto[]: Arreglo de enteros con el texto a cifrar o descifrar según sea el caso.
    - subKey1[] y subKey2[]: Subclaves que se generan a partir de la clave introducida por el usuario.

- S0 Y S1: Almacena los valores de las S-Boxes con las cuales se hará una posterior combinación.

Asimismo, dentro de esta clase se encuentran los siguientes métodos:

- DES(): Es el constructor de la clase. Se encarga de recibir la clave y el texto proporcionados por el usuario y acomodarlos como arreglos
- llenadoArreglos(): Recibe una cadena de texto (String) y la convierte a arreglo de enteros.
- imprimirArreglo(): Recibe un arreglo e imprime el contenido mediante un ciclo (y en una misma línea).
- permutacion(): Recibe un arreglo y lo permuta acorde a las posiciones indicadas en otro arreglo.
- division(): Recibe un arreglo y lo devuelve a la mitad, devolviendo la mitad que le sea indicada.
- desplazamiento(): Recibe un arreglo y lo mueve n posiciones según le sea indicado mediante el uso de un ciclo.
- union(): Recibe 2 arreglos y los une en uno solo.
- binario(): Recibe un valor entero y lo transforma a una cadena binaria.
- obtencionSubkeys(): Realiza la obtención de las subkeys acorde a la clave proporcionada por el usuario. Sus pasos generales incluyen permutación, división, desplazamiento, unión de cadenas, obtención de la subkey 1, desplazamiento, unión y obtención de la subkey 2.
- feistel(): Realiza la aplicación de la función Feistel a partir de la subclave que le sea indicada. En resumen, divide un arreglo inicial y a la parte de la derecha la expande y aplica xor con la subclave para su respectiva mezcla. A continuación, divide los datos y aplica la mezcla con las 2 S-Box declaradas para luego concatenar los resultados y permutarlos. Realiza un XOR con la mitad izquierda, concatena con la mitad derecha y obtiene el resultado final.
- cifrado(): Realiza los pasos que permiten el cifrado de un texto plano a partir del uso del algoritmo DES. En resumen, obtiene las subkeys, realiza una permutación inicial, aplica feistel con la Subkey 1, Intercambia los valores y aplica Fesitel con la subkey 2. Realiza una permutación final y devuelve el mensaje cifrado.
- descifrado(): Realiza los pasos que permiten el descifrado de un texto encriptado a partir del uso del algoritmo DES. En resumen, obtiene las subkeys, realiza una permutación inicial, aplica feistel con la Subkey 2, Intercambia los valores y aplica Fesitel con la subkey 1. Realiza una permutación final y devuelve el mensaje cifrado

Para la ejecución de este programa, se implementa un menú con las siguientes 3 opciones:

1. **Encrypt**: Al seleccionar esta opción, se realiza la encriptación de un mensaje que un usuario brinde junto con su clave correspondiente.

2. **Decrypt**: Al seleccionar esta opción, se realiza el descifrado de un mensaje dado a partir del algoritmo propuesto, haciendo uso de la clave y texto proporcionados por el usuario.
3. **Exit**: Termina la ejecución del programa.

Cabe destacar que el programa se repite de manera infinita hasta que el usuario seleccione la opción de "Exit".

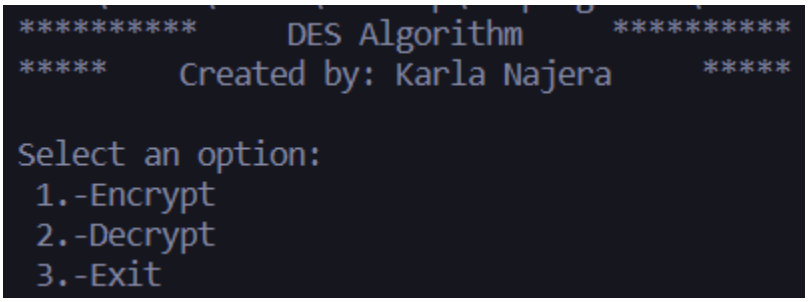
Aunque las clases ya han sido compiladas, en caso de ser requerido se puede compilar con la siguiente instrucción:

javac Main.java

Y se ejecuta con el siguiente comando:

java Main.java

Nota: Para su compilación es necesario tener instalada alguna versión de JDK, aunque la versión que fue utilizada para su elaboración es la 18.0.2.1.



A continuación, se muestran algunas capturas de pantalla de su funcionamiento tomando como referencia los vectores de prueba proporcionados para esta práctica

Llave y Texto Plano	Cifrado	Descifrado
0000011111 **01010101**	Key: 0000011111 Plaintext: 01010101 Ciphertext: 11000100	Key: 0000011111 Ciphertext: 11000100 Plaintext: 01010101
0000000000 **00000000**	Key: 0000000000 Plaintext: 00000000 Ciphertext: 11110000	Key: 0000000000 Ciphertext: 11110000 Plaintext: 00000000
1111111111 **11111111**	Key: 1111111111 Plaintext: 11111111 Ciphertext: 00001111	Key: 1111111111 Ciphertext: 00001111 Plaintext: 11111111

Aunque se adjunta un archivo en zip con los archivos utilizados para esta práctica, se comparte el enlace de GitHub: https://github.com/kiliapplered/Practica3Cripto.git

Conclusiones

Del presente trabajo es posible obtener las siguientes conclusiones:

- A partir de la presente práctica fue posible conocer uno de los algoritmos más famosos de cifrado simétrico, siendo en este caso el algoritmo DES en su versión simplificada.
- La hipótesis presentada para esta práctica fue correcta, siendo que se utilizaron los elementos básicos de los algoritmos simétricos, siendo en específico el uso de la función Feistel y la generación de subclaves a partir de los datos brindados por un usuario.
- Aunque hoy en día podría resultar simple el funcionamiento de este algoritmo e incluso cuestionable debido a que en la actualidad se encuentra "roto", no quedan dudas que en su momento presentó un avance significativo con respecto a lo que hoy conocemos como Criptografía Moderna.
- La implementación de este algoritmo como una solución "compleja y segura" no es deseable en estos días. En su lugar debe optarse por soluciones con un mayor nivel de complejidad y de los cuales no se conozca el algoritmo (o este sea mucho más difícil de descifrar). La alternativa más segura en este caso es AES, el cual no presenta una ruptura el su funcionamiento hasta el momento y se le considera la actualización de DES.
- La realización de este ejercicio permitió solidificar los conocimientos adquiridos durante la revisión teórica de la asignatura.

Referencias

[1] J. J. Meneu, «Detalles sobre la criptografía moderna,» Arrow, 18 Mayo 2016. [En línea]. Available: https://www.arrow.com/es-mx/research-and-events/articles/modern-cryptography. [Último acceso: 20 Septiembre 2022].

[2] D. E. Bernal Michelena, «LA CRIPTOGRAFÍA: EL SECRETO DE LAS COMUNICACIONES SEGURAS,» *Revista Seguridad UNAM,* nº 11, 2016.

[3] Geeks for Geeks, «Simplified Data Encryption Standard Key Generation,» Geeks for Geeks, 27 Septiembre 2021. [En línea]. Available: https://www.geeksforgeeks.org/simplified-data-encryption-standard-key-generation/. [Último acceso: 27 Septiembre 2022].

[4] K. Baloch, «S-DES or Simplified Data Encryption Standard,» C#Corner, 9 Marzo 2016. [En línea]. Available: https://www.c-sharpcorner.com/article/s-des-or-simplified-data-encryption-standard/. [Último acceso: 27 Septiembre 2022].

[5] «What is the Simplified Data Encryption Standard?,» TutorialsPoint, 17 Noviembre 2021. [En línea]. Available: https://www.tutorialspoint.com/what-is-the-simplified-data-encryption-standard. [Último acceso: 28 Septiembre 2022].

[6] Geeks for Geeks, «Simplified Data Encryption Standard | Set 2,» Geeks for Geeks, 22 Octubre 2021. [En línea]. Available: https://www.geeksforgeeks.org/simplified-data-encryption-

standard-set-2/#:~:text=Simplified%20Data%20Encryption%20Standard%20is,takes%2064%2Dbit%20plain%20text.. [Último acceso: 28 Septiembre 2022].

Yo, Karla Andrea Najera Noyola, hago mención que esta práctica fue de mi autoría.