**UNIVERSIDAD DE SANTIAGO DE CHILE**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

Sistemas de Comunicación

“Laboratorio 3: Criptografía”

Profesor:

Ayudantes:

Preparado por:

Carlos González Cortés

Catalina Morales

Hugo Arenas  
Victor Huanqui

**Tabla de Contenidos**

[1 Introducción 4](#_Toc62177410)

[2 Marco Teórico 5](#_Toc62177411)

[2.1 Criptografía Simétrica 5](#_Toc62177412)

[2.2 Cifrado por Bloques 5](#_Toc62177413)

[3 Desarrollo y Resultados 6](#_Toc62177414)

[4 Análisis de Resultados 9](#_Toc62177415)

[5 Conclusiones 10](#_Toc62177416)

[6 Referencias 11](#_Toc62177417)

# Introducción

La criptografía es una práctica que se ha realizado desde al menos 500 años a.C. Ha estado presente en el arte, las ciencias, las guerras y en los últimos años en la informática. Enviar un mensaje de forma segura de un lugar a otro ha sido una tarea difícil de completar, pero necesaria para el mundo desde que existe la comunicación, sobre todo cuando se trata de información confidencial. Es por este motivo que tanto la encriptación de mensajes como el criptoanálisis han sido temas tan relevantes durante periodos tan cruciales como las guerras mundiales. Actualmente, en un mundo donde dominan las computadoras, en donde todo proceso es realizado a través de internet (pagos, postulaciones, compras, servicios, ventas, etc.), es de vital importancia que los datos que se comparten de forma privada se mantengan así desde su origen hasta el destino.

A lo largo de la historia han existido múltiples formas de encriptar mensajes, pero centrándose en los tiempos actuales, la criptografía moderna es lo que rige al mundo actualmente. Es por este motivo que en el presente informe se abordará el tema de la criptografía simétrica por medio del diseño de un sistema de encriptación que no sea susceptible a ciber ataques. El sistema de cifrado tiene que especificar ciertos parámetros como el tamaño de la llave, tamaño del mensaje, tamaño de los bloques, función de encriptación y función de desencriptación.

# Marco Teórico

## Criptografía Simétrica

La criptografía de clave simétrica o también llamada criptografía de clave secreta, es un método criptográfico en el cuál se usa la misma clave para cifrar y descifrar mensajes en el emisor y receptor. En este proceso, ambas partes deben ponerse de acuerdo para conocer de antemano la clave a usar.

## Cifrado por Bloques

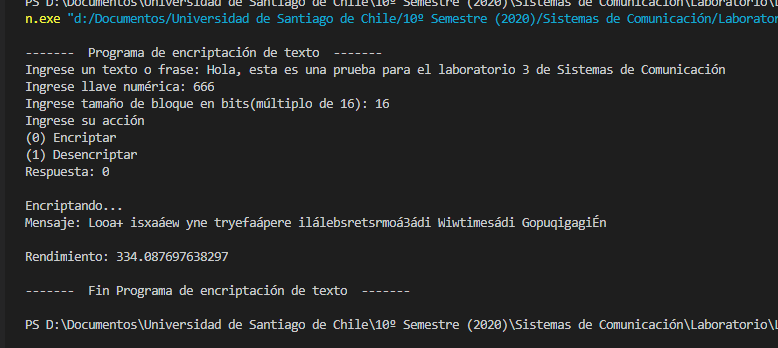
Se realiza sobre cifrados simétricos y es una unida que opera con grupos de bits de longitud fija llamados bloques. El proceso consta en aplicar una transformación invariable a un bloque de texto plano que tiene como resultado un bloque del mismo tamaño al de entrada. La transformación depende de la llave de entrada y el descifrado es similar, produce un bloque de texto plano.

# Desarrollo y Resultados

Para los puntos 1, 2 y gran parte del punto 3 del enunciado se encuentran en el código.

Para el desarrollo del sistema de encriptación se usaron los siguientes parámetros:

* Lenguaje: “0123456789abcdefghijklmnñopqrstuvwxyzABCDEFGHIJKLMNÑOPQRSTU VWXYZ#$%&/\{}[]()-~\_,.;:+\*'= °|¬áéíóúüÁÉÍÓÚÜ´âêîôûÂÊÎÔÛ^àèìòùÀÈÌÒÙ`¿?¡!<>@”.
* La llave es del tipo numérica y debe ser un número entero mayor que cero.
* El largo del bloque es de 16n bits con n entero positivo escogido por el usuario.



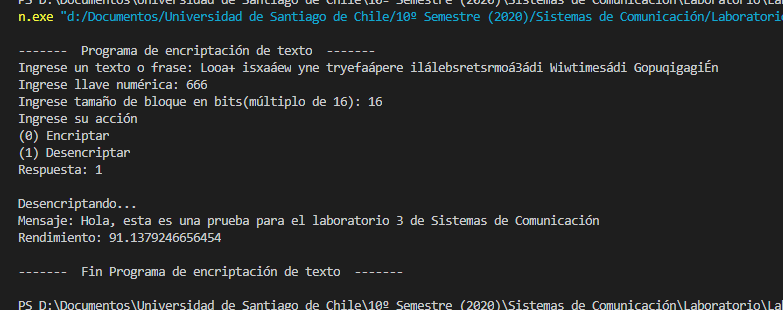
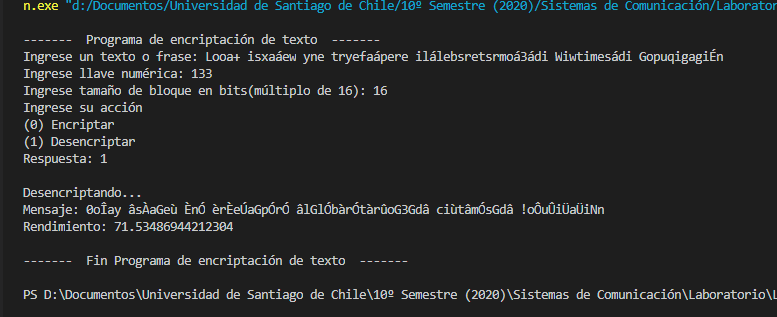
Figura 3-1: Ejemplo encriptación de un mensaje.

Figura 3-2: Ejemplo desencriptación de un mensaje con llave correcta.

Figura 3-3: Ejemplo desencriptación de un mensaje con llave incorrecta.

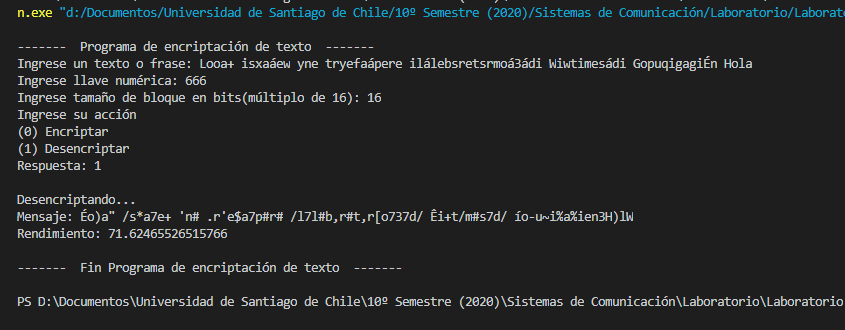
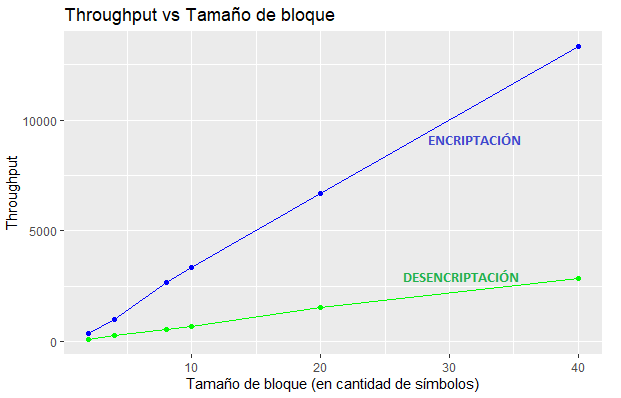
Para mostrar el efecto avalancha, se agregará un “ Hola” al final del texto encriptado, se usará la misma clave y tamaño de bloque.

Figura 3-4: Ejemplo efecto avalancha.

Gráfico 3-1: Rendimiento del programa dependiendo del tamaño del bloque en la encriptación y desencriptación.

# Análisis de Resultados

Para la encriptación del mensaje se recorre el texto dependiendo de el largo del bloque y los símbolos que contenga, el proceso se divide en dos partes, el primer fragmento y el segundo. Al primer fragmento se aplica una sustitución con la llave de entrada y luego se realiza una permutación con el segundo fragmento. Este proceso se repite iterativamente la cantidad de veces necesaria dependiendo del tamaño del mensaje ingresado.

Como se muestra en la *Figura 3-1*, se ingresa un mensaje con llave 666 y tamaño del bloque 16bits. Como resultado se obtiene un mensaje cifrado del mismo largo que el mensaje de entrada.

En la *Figura 3-2*, se muestra el desencriptado del mensaje ingresado originalmente, obteniendo como resultado el mensaje ingresado inicialmente, valga la redundancia. Se ingresa la misma clave y tamaño de bloque inicial para el desencriptado obteniendo como salida el mensaje original.

En la *Figura 3-3*, se muestra una entrada incorrecta de la llave, el resultado es completamente diferente a la entrada inicial y no tiene ningún parecido con el texto original. Esto prueba que hacer el uso de una llave incorrecta conlleva a la deformación completa del mensaje original.

En la *Figura 3-4*, se muestra el efecto avalancha, agregando una cadena de caracteres de solo 5 símbolos el mensaje se pierde completamente a pesar de tener una entrada de llave correcta. Esto prueba que, a pequeñas modificaciones del mensaje original, este es eliminado completamente.

En el *Gráfico 3-1*, se muestra que el rendimiento del programa. Como se aprecia en la imagen tiempo de ejecución en la encriptación y desencriptación aumentan de forma lineal. Sin embargo, con distinta pendiente. La encriptación en términos de ejecución es mucho más costosa. Por otro lado, cada ejecución del programa entrega un rendimiento como se puede apreciar en las figuras mencionadas anteriormente, cada una entrega el rendimiento que concuerda con los resultados del gráfico.

El diseño del algoritmo para la encriptación y desencriptación tiene como base el cifrado Feistel. Sus similitudes son el uso de una función de sustitución por fragmento y la permutación entre el fragmento sustituido y el otro fragmento del mismo bloque. También se realiza el cifrado se repite una N cantidad de veces, donde N es igual al largo del texto a encriptar/desencriptar. Al igual que Feistel, también se utiliza una llave generada para realizar la sustitución. Las diferencia que tiene el encriptador/desencriptador aplicado en este laboratorio con respecto al cifrado Feistel es que solo utiliza una sola llave que es dada por el usuario para realizar la sustitución, donde esta llave hace la función de moverse unas L posiciones a la derecha o izquierda, dependiendo si es encriptación o desencriptación, desplazándose por una lista hash que posee todo los símbolos disponibles. Con respecto a la permutación, para la encriptación, va agregando a un nuevo fragmento, de forma paralela, desde el primer símbolo desde el primer fragmento y desde el último símbolo del segundo fragmento. Por ejemplo, si se tiene el primer fragmento “ABC” y el segundo fragmento “DEF”, entonces el nuevo fragmento es “AFBECD”. Para el caso de la permutación en la desencriptación, las permutación se realiza por sobre el fragmento entero del bloque, creando un primer fragmento tomando desde el primer símbolo del fragmento entero hacia 2 posiciones a la derecha hasta llegar al final. El segundo fragmento se crea tomando desde el último símbolo del fragmento entero hacia e posiciones a la izquierda hasta llegar al inicio. Por ejemplo, si el fragmento entero es “ABCDEF”, entonces el primer fragmento es “ACE” y el segundo fragmento es “FDB”.

La principal ventaja del cifrado implementado con respecto al cifrado Feistel es que es menos propenso a los ataques tipo *chosen plaintext* en casos particulares. Por ejemplo, solo es más vulnerable a estos ataques cuando la llave es 0 y las repeticiones para el cifrado es (N - 1) o múltiplo de este, pues se llegaría al mismo texto. Además, solo requiere de una sola llave y no generar distintas llaves por cada repetición. También puede ajustar el tamaño del texto si su largo es impar agregando un espacio al final, como también puede ajustar el tamaño del bloque para que sea divisor del largo del texto y conservando que sean par.

Sin embargo, su desventaja principal con respecto al cifrado Feistel es que puede llegar a demorar más tiempo con respecto al tema de la permutación y la operación de desencriptación. Esto se debe principalmente a que su mécanica es más compleja a la hora de realizar la permutación y las operaciones necesarias para desencriptar. Esto también depende de cuando demore el cifrado Feistel en originar las llaves a utilizar y de las funciones de sustitución que utilicen cada uno.

# Conclusiones

El sistema implementado se basó en una encriptación simétrica, quiere decir, con una misma llave tanto para la encriptación como desencriptación y un tamaño de ingreso y salida del mensaje igual. El desarrollo del programa fue pensado para un uso del lenguaje en español ya que contiene todos sus símbolos. Además, la llave utilizada debe se de tipo entera y positiva. En los ejemplos se muestra con claridad la ejecución de la aplicación, tanto en sus entradas como salidas, se aprecian la encriptación tanto como desencriptación exitosa y fallida.

El uso de este tipo de sistema de cifrado de datos es de vital importancia en todo lo relevante a la informática, ya que en todo momento existen traspasos de información de un servidor a otro y estos bloques de información deben ir protegidos de tal manera que no sea posible, para otra persona que no tiene la llave, entender su contenido.

A lo largo de la experiencia se puedo crear un algoritmo capaz de encriptar un mensaje de forma exitosa a partir de una entrada de tipo string, una llave entera y un tamaño de bloque definido por el usuario. Por otro lado, es importante considerar la eficiencia del algoritmo, ya que si se debe usar en sistemas en donde la repetición del cifrado sea importante, pueden existir problemas de rendimiento, no es el caso de este, ya que cumple con el propósito planeado.

# Referencias