

Secret Safe

**Santiago Tovar Mosquera**

**Juliana Alejandra Nieto Cárdenas**

**Anderson Andrés Llanos Quintero**

**Daniel Estevan Garcia Niño**

2023

Introducción a la Criptografía y a la teoría de la Información

Departamento de Matemáticas

Universidad Nacional de Colombia, Sede Bogotá

[**Introducción 4**](#_15316qh1jyb8)

[Objetivos de la aplicación 4](#_5q4q8bebe10e)

[Objetivo principal 4](#_23j3u9n7a6od)

[Objetivos secundarios 4](#_h123jzgcow01)

[Usuarios objetivo 5](#_fi1bn16l1km1)

[**Instalación 6**](#_fpp20fgvagkt)

[Requisitos del sistema 6](#_swf08zob227y)

[Proceso de instalación 6](#_v98svldvngt6)

[Configuración inicial 7](#_nxvjuziwgfj5)

[**Guía de usuario 8**](#_qjd5ccfsufw)

[Primeros pasos 8](#_snqgjrlfztle)

[Funcionalidades principales 8](#_ffgsgqatze5k)

[Criptografía clásica 9](#_olpltuenx9yw)

[Desplazamiento 9](#_a1ds3e3bdhep)

[Afín 9](#_xqida8u60xe4)

[Hill 9](#_yomofdfbjv26)

[Sustitución-Permutación 10](#_u9qgqrmc0d0i)

[Vigenère 10](#_cis10t8m7kwx)

[Criptografía de imágenes 10](#_xhxvx72g22jr)

[Hill 10](#_hpf10yckagp7)

[AES 11](#_enxmcctyw5qz)

[S DES 11](#_o703kiaugw2u)

[Triple-DES 12](#_28bdh93vpm4u)

[Criptografía de clave pública 13](#_y5z47b93aah4)

[ElGamal 13](#_u897dsojkznt)

[Menezes-Vanstone 13](#_1aji6fefwenu)

[Rabin 13](#_3k61yt5ymlj6)

[RSA 14](#_nyx2hvpwx89i)

[Criptoanálisis para Vigenère 14](#_49ju2xbadyfo)

[Ejemplos de uso 15](#_1a4bzj47rjh)

[Criptografía clásica 15](#_17hh955van3)

[Desplazamiento 15](#_9rsvgi4wtejq)

[Afín 16](#_d64tvaoq76fj)

[Hill 17](#_naog1yozt1y9)

[Sustitución-Permutación 18](#_oagasxrc7vxc)

[Vigenère 19](#_6yu72t58uafo)

[Criptografía de imágenes 20](#_fpvoff7pjwtf)

[Hill 20](#_br3nz3spslrh)

[AES 20](#_jbdw64f76e22)

[SDES 21](#_dn9k7d3fuzrr)

[Triple-DES 23](#_35nrz5387wk9)

[Criptografía de clave pública 24](#_ul01qohijlth)

[ElGamal 24](#_wy2khumctxh0)

[Menezes-Vanstone 24](#_8qvudad0ykoz)

[Rabin 26](#_abvmzp2keb9h)

[RSA 27](#_1v8ti03xunpf)

[Criptoanálisis para Vigenère 28](#_syxo7gt19xlv)

### 

# Introducción

La criptografía desempeña un papel fundamental en la era moderna al garantizar seguridad, integridad y autenticidad de la información en el ámbito digital. Mediante el cifrado, se protege la confidencialidad de datos sensibles, asegura la integridad de la información y facilita la autenticación. Además, la criptografía desempeña un papel clave en la protección contra ataques cibernéticos, preservando la privacidad individual.

Aunque la seguridad informática es un tema de suma relevancia en el mundo globalizado, en Colombia no se implementa de manera sistemática. Con frecuencia, las páginas web, documentos y conversaciones gubernamentales se convierten en blanco de ataques cibernéticos, generando así una creciente inseguridad a nivel nacional.

Sin embargo, no solo el gobierno está en riesgo; los ciudadanos colombianos en su vida diaria podrían perder información, ser víctimas de extorsiones y enfrentar otros problemas derivados de ataques a la información.

Es por esta razón que es crucial desarrollar una aplicación de criptografía, en orden de:

* Garantizar la seguridad, integridad y autenticidad de la información en el ámbito digital.
* Asegurar la privacidad en las comunicaciones.
* Contribuir al desarrollo tecnológico del país y avanzar en ciberseguridad.
* Democratizar las herramientas de seguridad entre los ciudadanos.

## Objetivos de la aplicación

### Objetivo principal

Desarrollar una herramienta versátil de encriptado y desencriptado que admita diversos sistemas criptográficos, además de incorporar funcionalidades para el criptoanálisis específico del cifrado Vigenère.

### 

### Objetivos secundarios

* Facilitar el cifrado de mensajes de texto e imágenes, dos medios comunes de comunicación en la actualidad.
* Implementar cifrados clásicos: desplazamiento, afín, Hill, sustitución-permutación y Vigenère.
* Implementar cifrados para imágenes: Hill, AES, SDES y Triple-DES en los diferentes modos CBC, CTR, ECB y OFB.
* Implementar cifrados de clave pública: ElGamal, Menezes-Vanstone, Rabin y RSA.
* Diseñar una interfaz de usuario que sea intuitiva, simple y funcional, con el objetivo de proporcionar a los usuarios una experiencia accesible y eficiente.

## Usuarios objetivo

SecretSafe es una aplicación diseñada para cualquier individuo que forme parte de una red de comunicaciones, no exclusivamente en Internet, y que tenga acceso a una computadora con sistemas operativos ampliamente utilizados, como Windows, MacOS o Linux. Está destinada a aquellos que buscan una comunicación segura y confiable.

# 

# Instalación

## Requisitos del sistema

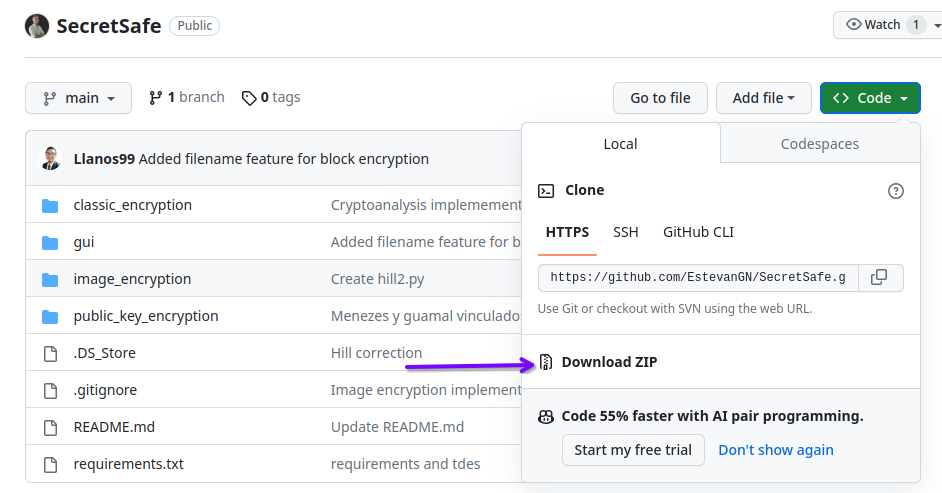
* Sistema Operativo: Compatible con MacOS, Windows, y Linux.
* Python: Se necesita Python versión 3.12.
* RAM: Se requieren al menos 2GB.
* Arquitectura: Debe ser de 64 bits.
* CPU: Se recomienda una CPU con una velocidad de 3.2 GHz.
* Espacio en Disco Duro: Se necesita al menos 1GB de espacio disponible.

## Proceso de instalación

La aplicación de SecretSafe se encuentra alojada en el siguiente repositorio remoto de GItHub

<https://github.com/EstevanGN/SecretSafe>

Para instalar la aplicación es necesario:

1. **Descargar el archivo ZIP del repositorio:** este contiene todo lo necesario para ejecutar SecretSafe  
   
2. **Crear un ambiente virtual:** Esto se hace para que las dependencias de la aplicación puedan trabajar en un ambiente aislado y seguro, evitando conflictos con otras dependencias. Para crear el ambiente virtual, primero hacemos cd gui y luego ejecutamos:  
   python -m venv gui/myenv  
   Y para activar el ambiente virtual, se usa el siguiente comando para MacOS y Linux:  
   source myenv/bin/activate  
   Para Windows se usa:  
   myenv\Scripts\activate.bat
3. **Instalar dependencias:** estos son componentes de Python que son usados por SecretSafe para su correcto funcionamiento. Más adelante, se explica a detalle cuáles son estas dependencias y para que son usadas. La lista de todas estas dependencias está en el archivo de texto *requirements.txt*, y pueden ser instaladas de la siguiente forma:  
   pip install -r requirements.txt
4. **Ejecutar la aplicación:** Accediendo a la carpeta gui se puede ejecutar la aplicación mediante el siguiente comando en terminal:  
   python main.py

## Configuración inicial

Para que SecretSafe funcione correctamente se necesitan las siguientes dependencias de Python.

* mpmath==1.3.0: Es una biblioteca de Python para matemáticas de precisión arbitraria. Permite realizar cálculos con números de punto flotante de alta precisión.
* Nuitka==1.8.6: Es un compilador de Python que traduce los scripts de Python a código de máquina, mejorando el rendimiento de la aplicación.
* numpy==1.26.2: Proporciona soporte para grandes matrices y matrices multidimensionales junto con una colección de funciones matemáticas para trabajar con estas matrices.
* ordered-set==4.1.0: Implementa un tipo de datos que es una mezcla entre un conjunto (set) y una lista, manteniendo el orden de los elementos, pero asegurando que no haya duplicados.
* Pillow==10.1.0: Es una biblioteca de manipulación de imágenes que añade soporte para abrir, manipular y guardar muchos formatos de archivos de imágenes diferentes.
* pycryptodome==3.19.0: Ofrece criptografía segura, proporcionando herramientas para trabajar con cifrado simétrico, criptografía de clave pública, y más.
* PySide6==6.6.0: Es un conjunto de enlaces Python para Qt, un framework para desarrollar interfaces gráficas de usuario y otras aplicaciones que requieren elementos visuales.
* PySide6-Addons==6.6.0: Proporciona complementos adicionales para PySide6, mejorando sus capacidades para el desarrollo de interfaces gráficas.
* PySide6-Essentials==6.6.0: Incluye los componentes esenciales de PySide6 necesarios para el desarrollo básico de aplicaciones con interfaces gráficas.
* shiboken6==6.6.0: Genera enlaces entre C++ y Python, lo que permite la integración de bibliotecas C++ en aplicaciones Python.
* sympy==1.12: Es una biblioteca de Python para matemáticas simbólicas. Permite a los usuarios realizar cálculos algebraicos y matemáticos sin necesidad de realizar operaciones numéricas.
* zstandard==0.22.0: Es una biblioteca de compresión, proporcionando una implementación de la codificación de compresión Zstandard, que se destaca por su eficiencia y velocidad.

# Guía de usuario

## Primeros pasos

Después de ejecutar la aplicación mediante el comando descrito anteriormente, se verá en pantalla la ventana principal de SecretSafe. Como cualquier otra aplicación de escritorio, SecretSafe se puede minimizar, maximizar y cerrar.



## Funcionalidades principales

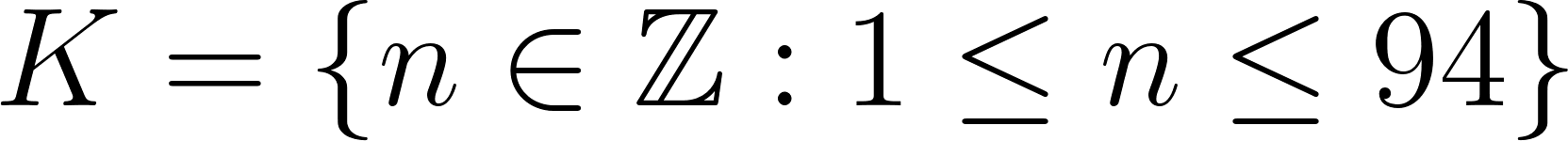
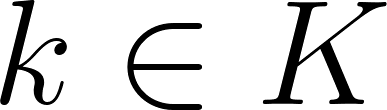
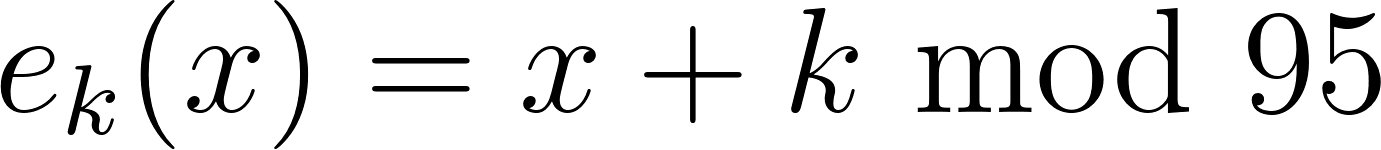
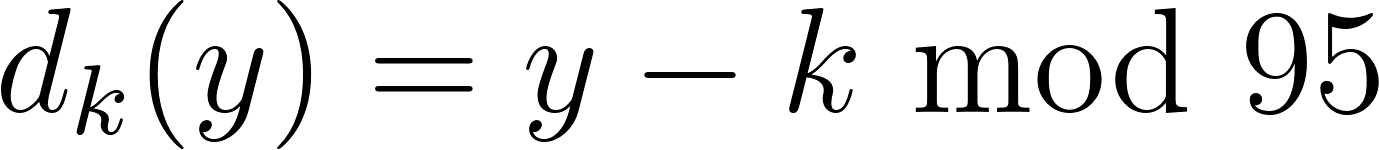
SecretSafe presenta a la izquierda un menú principal con cuatro funciones distintas. Tres de ellas están dedicadas a la criptografía: 'Criptografía Clásica', 'Criptografía de Imágenes', y 'Criptografía de Clave Pública', cada una enfocada en diferentes aspectos del cifrado y descifrado de información. La cuarta opción es una herramienta especializada para el criptoanálisis de Vigenère, ampliando así las capacidades de la aplicación en el campo de la seguridad informática.

### Criptografía clásica

La criptografía clásica abarca los métodos antiguos de cifrado de mensajes utilizados antes de la era de la computación. Estos métodos, en su mayoría manuales, se enfocan en transformar el texto original (texto plano) en un formato codificado (texto cifrado) para proteger su contenido durante la transmisión. SecretSafe implementa cifrado por desplazamiento, cifrado por sustitución, cifrado Vigenère, cifrado afín y cifrado de Hill, para cifrar mensajes de texto.

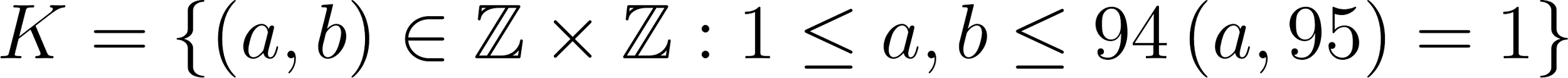
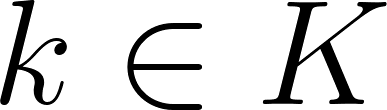
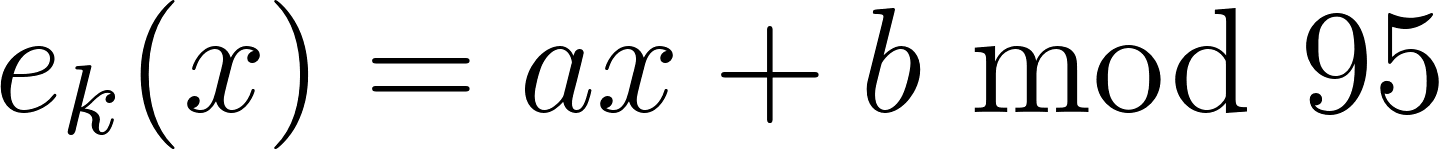
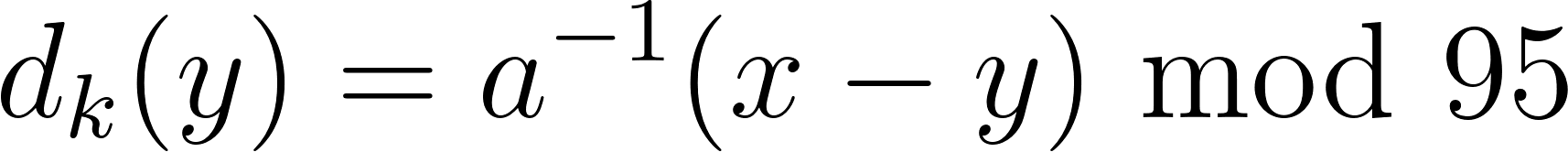
#### Desplazamiento

El cifrado por desplazamiento, también conocido como cifrado César, es un tipo de cifrado de sustitución en el que cada letra en el texto plano se "desplaza" un cierto número de posiciones hacia abajo o hacia arriba en el alfabeto. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano y cifrado, caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20C%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de claves [](https://www.codecogs.com/eqnedit.php?latex=K%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%201%20%5Cleq%20n%20%5Cleq%2094%20%5C%7D#0)
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3D%20x%2Bk%20%5Ctext%7B%20mod%20%7D%2095#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20y%20-%20k%20%5Ctext%7B%20mod%20%7D%2095#0)

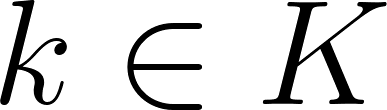
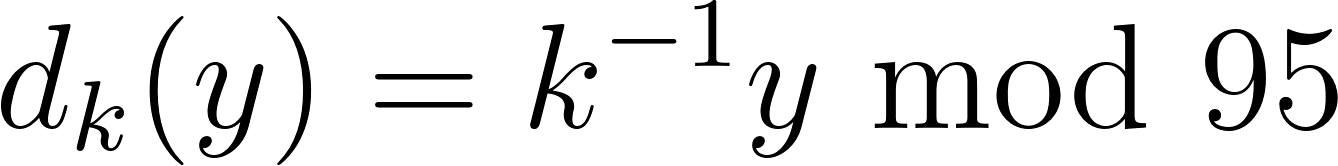
#### Afín

El cifrado afín es un tipo de cifrado de sustitución que combina dos operaciones matemáticas, la multiplicación y la suma, para cifrar un mensaje. Este método se basa en la aritmética modular y utiliza dos claves: una para la multiplicación y otra para la suma. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano y cifrado, caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20C%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de claves [](https://www.codecogs.com/eqnedit.php?latex=K%20%3D%20%5C%7B(a%2Cb)%20%5Cin%20%5Cmathbb%7BZ%7D%5Ctimes%5Cmathbb%7BZ%7D%20%3A%201%20%5Cleq%20a%2Cb%20%5Cleq%2094%20%5C%2C(a%2C95)%3D1%5C%7D#0)
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3D%20ax%2Bb%20%5Ctext%7B%20mod%20%7D%2095#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20a%5E%7B-1%7D(x-y)%20%20%5Ctext%7B%20mod%20%7D%2095#0)

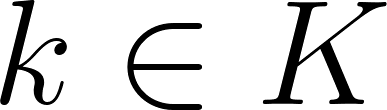
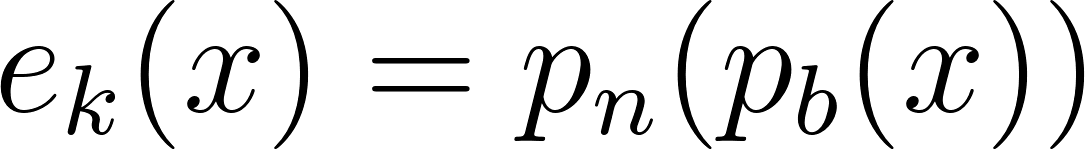
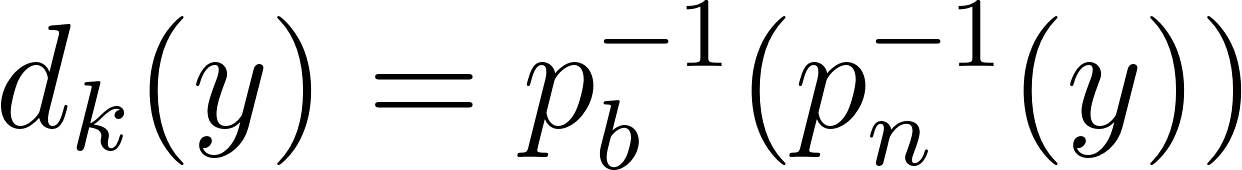
#### Hill

El cifrado de Hill es un algoritmo de cifrado de sustitución poligráfico que se basa en el uso de álgebra lineal, particularmente en la teoría de matrices. En este cifrado, el texto se divide en bloques de tamaño *n* (donde *n* es un número entero), y cada bloque de *n* letras se trata como un vector. Este vector se multiplica entonces por una matriz *n×n* que actúa como la "clave" del cifrado. La clave debe ser elegida de tal manera que sea invertible en aritmética modular. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano y cifrado, caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20C%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de claves [](https://www.codecogs.com/eqnedit.php?latex=K%20%3D%20%5Cleft%5C%7B%20A%20%5Cin%20%5Cmathbb%7BM%7D_%7B4%20%5Ctimes%204%7D(%5Cmathbb%7BZ%7D)%20%5Cmid%20%5Cdet(A)%20%5Cneq%200%2C%20%5Ctext%7B%20y%20todas%20las%20entradas%20%7D%20a_%7Bij%7D%20%5Ctext%7B%20de%20%7D%20A%20%5Ctext%7B%20satisfacen%20%7D%2032%20%5Cleq%20a_%7Bij%7D%20%5Cleq%20126%2C%5C%2C%20%5Cleft(%5Cdet(A)%2C95%5Cright)%3D1%20%5Cright%5C%7D#0)
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3Dkx%20%5Ctext%7B%20mod%20%7D%2095#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20k%5E%7B-1%7Dy%20%20%5Ctext%7B%20mod%20%7D%2095#0).

#### Sustitución-Permutación

El cifrado por sustitución-permutación es un método de cifrado donde cada elemento del texto plano (como una letra o un grupo de letras) se reemplaza sistemáticamente por otro elemento. Este método es uno de los más antiguos y conocidos en criptografía. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano y cifrado, caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20C%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* Según el texto a cifrar, se genera una clave, la cual consiste en un conjunto de permutaciones pequeñas [](https://www.codecogs.com/eqnedit.php?latex=p_b#0) que dividen el texto en bloque, y una permutación del tamaño del texto [](https://www.codecogs.com/eqnedit.php?latex=p_n#0).
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3Dp_n(p_b(x))#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20p_%7Bb%7D%5E%7B-1%7D(p_%7Bn%7D%5E%7B-1%7D(y))#0)

#### Vigenère

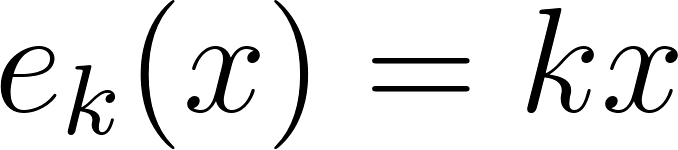
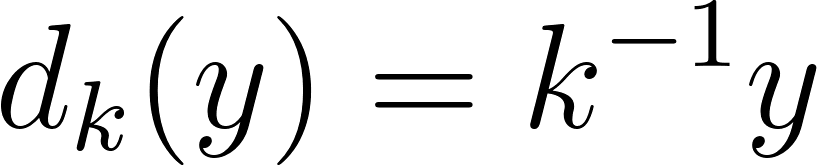
El cifrado Vigenère es un método de cifrado de texto que utiliza una serie de diferentes cifrados César basados en las letras de una palabra clave. Es un ejemplo simple de un cifrado polialfabético, lo que significa que utiliza múltiples alfabetos para cifrar el texto. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano y cifrado, caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20C%20%3D%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* La clave es también un elemento del conjunto de texto plano y cifrado.
* El proceso de cifrado consiste en sumar letra a letra el texto plano con la clave. El proceso de descifrado sería la operación inversa al cifrado que sería restar.

### Criptografía de imágenes

#### Hill

El cifrado de Hill, originalmente diseñado para textos, también se puede adaptar para cifrar imágenes. En el contexto de las imágenes, este método utiliza conceptos de álgebra lineal para transformar los píxeles de la imagen. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano son imágenes en formato PNG o JPG y el conjunto de texto cifrado son imágenes en formato PNG.
* El conjunto de claves [](https://www.codecogs.com/eqnedit.php?latex=K%20%3D%20%5C%7B%20A%20%5Cin%20%5Cmathbb%7BM%7D_%7B3%20%5Ctimes%203%7D(%5Cmathbb%7BZ%7D)%20%5Cmid%20%5Cdet(A)%20%5Cneq%200%20%5Ctext%7B%20y%20todas%20las%20entradas%20%7D%20a_%7Bij%7D%20%5Ctext%7B%20de%20%7D%20A%20%5Ctext%7B%20satisfacen%20%7D%201%20%5Cleq%20a_%7Bij%7D%20%5Cleq%20256%2C%5C%2C%20%5Cleft(%5Cdet(A)%2C%20256%5Cright)%3D1%20%5C%7D#0)
* La función de cifrado es [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3D%20kx#0) y la función de descifrado es [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20k%5E%7B-1%7Dy#0).

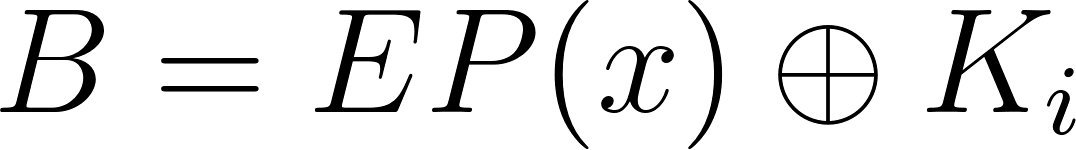
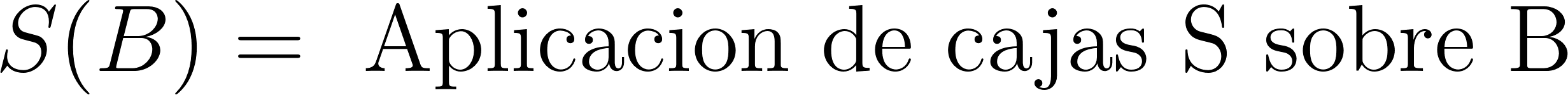
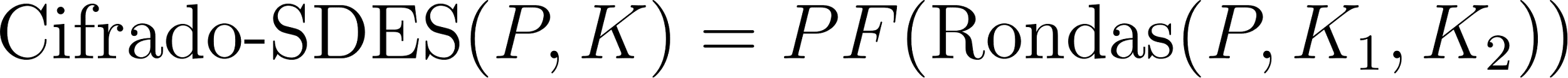
#### AES

El Estándar de Cifrado Avanzado (AES, por sus siglas en inglés) es un algoritmo de cifrado de bloque ampliamente utilizado, que también se puede adaptar para cifrar imágenes. En el contexto de las imágenes, AES utiliza una serie de transformaciones criptográficas para modificar los píxeles de la imagen. Este criptosistema se define de la siguiente manera:

* El conjunto de texto plano y cifrado son imágenes en formato PNG Cada imagen se trata como una serie de bloques de datos, donde cada bloque contiene un conjunto específico de píxeles.
* AES utiliza una clave simétrica, que puede ser de 128, 192 o 256 bits. Esta clave es utilizada tanto para el cifrado como para el descifrado. La elección de la longitud de la clave determina el número de rondas de transformación que se aplicarán durante el cifrado y el descifrado.
* La imagen se divide en bloques de 128 bits, y cada bloque se cifra mediante una serie de rondas que incluyen sustitución (a través de una caja-S), permutación (desplazando filas), mezclado de columnas y una adición de la clave de ronda. La función de cifrado para un bloque se puede expresar como:  
  [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3D%20%5Ctext%7B%20Rondas%20de%20Transformaciones%20AES%7D(X%2CK)#0)  
  El proceso de descifrado aplica las rondas de transformación en orden inverso utilizando la misma clave

#### S DES

El S-DES (Simplified Data Encryption Standard) es una versión simplificada del algoritmo de cifrado DES (Data Encryption Standard). Fue diseñado con fines educativos para ayudar a los estudiantes a comprender los conceptos básicos de los algoritmos de cifrado más complejos como el DES. A pesar de ser una simplificación, S-DES conserva las características esenciales y la estructura general de DES, aunque en una escala mucho menor. Este criptosistema se define de la siguiente manera:

* El conjunto de texto plano y cifrado son imágenes en formato PNG Cada imagen se trata como una serie de bloques de datos, donde cada bloque contiene un conjunto específico de píxeles.
* S-DES utiliza una clave de 10 bits, que es mucho más corta que la clave de 56 bits usada en DES. A partir de esta clave, se generan dos subclaves de 8 bits cada una mediante un proceso de permutación y desplazamiento.
* Para ser cifrado, el texto plano se procesa en bloques de 8 bits. Cada bloque pasa por varias rondas de procesamiento, similares a las rondas de DES pero en menor cantidad.
  + Expansión/Permutación: Expande y permuta un bloque de 4 bits a 8 bits  
    [](http://www.sciweavers.org/tex2img.php?bc=Transparent&fc=Black&im=jpg&fs=100&ff=modern&edit=0&eq=EP(x)%20%3D%20%5Ctext%7BPermutacion%20de%20expansi%C3%B3n%20de%20x%7D#0)
  + Combinación con la subclave:  
    [](https://www.codecogs.com/eqnedit.php?latex=B%20%3D%20EP(x)%20%5Coplus%20K_i#0)
  + Caja S:  
    [](https://www.codecogs.com/eqnedit.php?latex=S(B)%20%3D%20%5Ctext%7B%20Aplicacion%20de%20cajas%20S%20sobre%20B%7D#0)
  + Permutación P4:  
    [](https://www.codecogs.com/eqnedit.php?latex=P4(y)%20%3D%20%5Ctext%7B%20Permutacion%20P4%20de%20y%7D#0)
  + Combinación con la Mitad del Bloque:  
    [](https://www.codecogs.com/eqnedit.php?latex=C%20%3D%20P4(y)%20%5Coplus%20%5Ctext%7B%20Mitad%20del%20bloque%20de%20P%7D#0)
  + [Intercambio de la](https://www.codecogs.com/eqnedit.php?latex=C%20%3D%20P4(y)%20%5Coplus%20%5Ctext%7B%20Mitad%20del%20bloque%20de%20P%7D#0)s dos mitades de P
  + Permutación final [](https://www.codecogs.com/eqnedit.php?latex=PF#0) del bloque procesado para obtener el texto cifrado
* De esta manera la función de cifrado se define como:  
  [](https://www.codecogs.com/eqnedit.php?latex=%5Ctext%7BCifrado-SDES%7D(P%2CK)%20%3D%20PF(%5Ctext%7BRondas%7D(P%2CK_1%2CK_2))#0)
* El proceso de descifrado, es el proceso de descifrado a la inversa.

#### Triple-DES

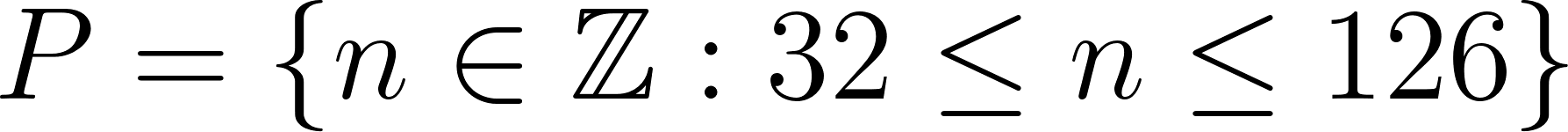
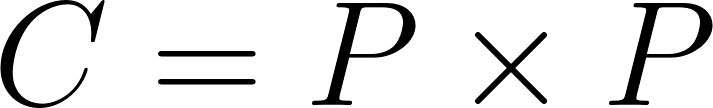
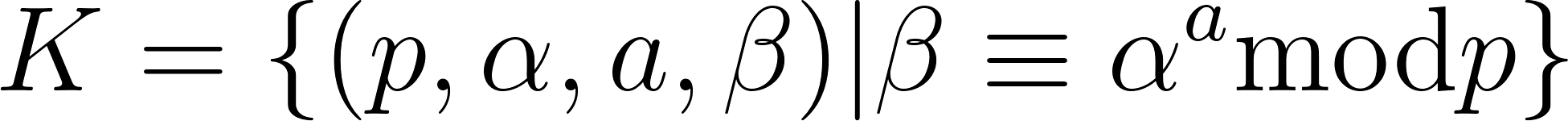
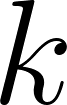
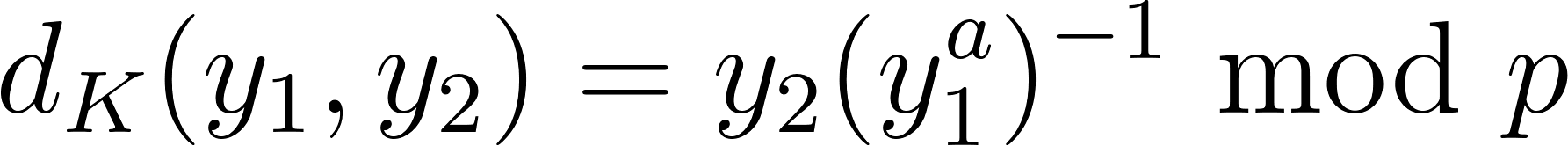
Triple DES (3DES) es una variante del Estándar de Cifrado de Datos (DES) que incrementa la seguridad del DES original mediante la aplicación de tres rondas de cifrado DES en secuencia. La idea detrás de 3DES es mejorar la seguridad del DES, que empezó a considerarse vulnerable debido a su clave relativamente corta de 56 bits. Este criptosistema se define de la siguiente manera:

* El conjunto de texto plano y cifrado son imágenes en formato PNG Cada imagen se trata como una serie de bloques de datos, donde cada bloque contiene un conjunto específico de píxeles.
* 3DES utiliza tres claves de cifrado DES, cada una de 56 bits, lo que da un total de 168 bits. Sin embargo, en la versión más común de 3DES (llamada 3DES EDE - Encriptar-Descifrar-Encriptar), solo se usan efectivamente 112 bits debido a que la primera y la tercera clave son iguales.
* Un bloque de texto plano se somete a tres fases sucesivas de cifrado y descifrado DES:
  + Primero, el bloque de texto plano se cifra con la primera clave (K1).
  + Luego, el resultado se descifra con la segunda clave (K2). Aunque esto parece contraproducente, el uso de una operación de descifrado con una clave diferente añade una capa adicional de seguridad.
  + Finalmente, el resultado se vuelve a cifrar con la tercera clave (K3, que es la misma que K1 en el esquema 3DES EDE).
* El descifrado de un bloque de texto cifrado con 3DES es el inverso del proceso de cifrado:
  + Primero, el bloque cifrado se descifra con la clave K3.
  + Luego, se cifra con la clave K2.
  + Finalmente, se descifra de nuevo con la clave K1.

### Criptografía de clave pública

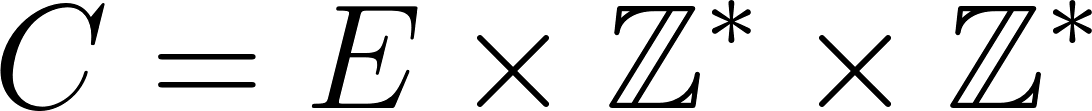
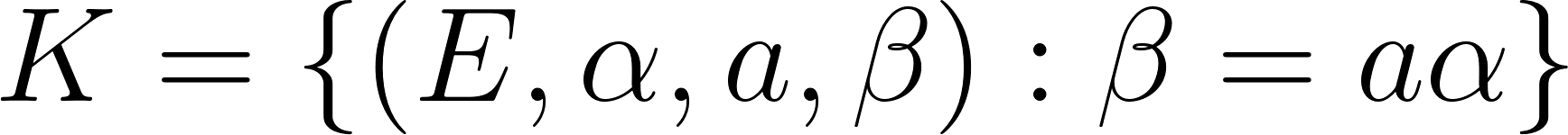
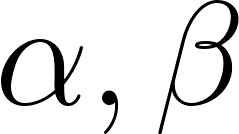
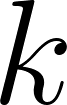
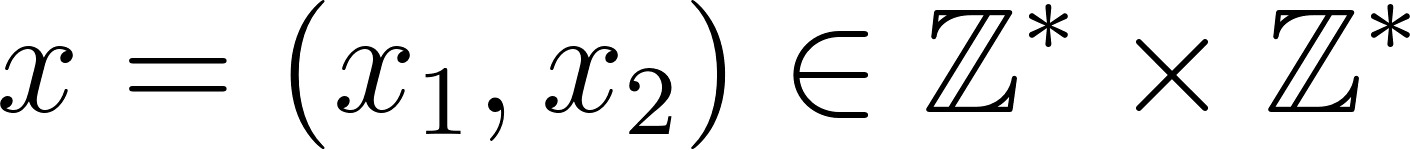
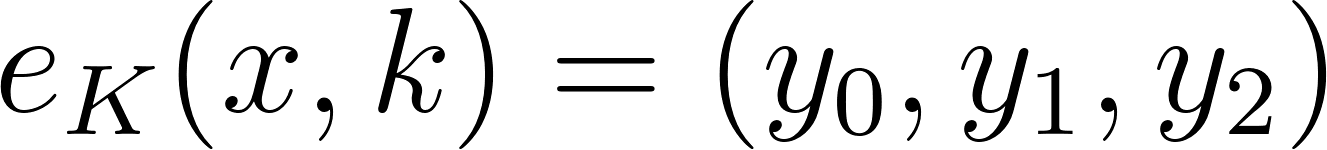
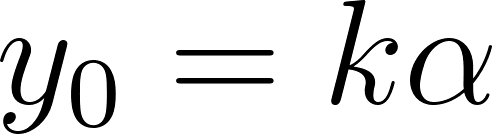
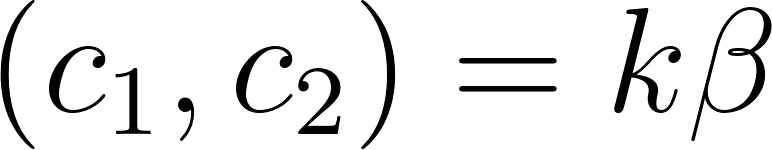
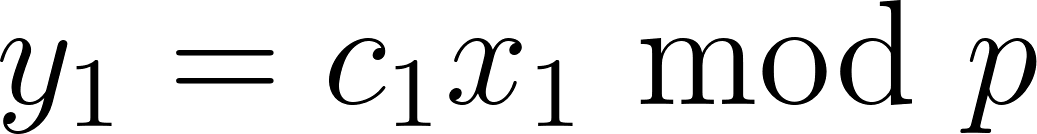
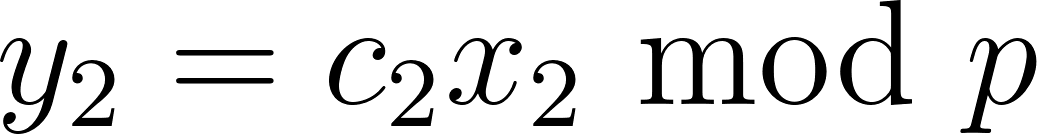
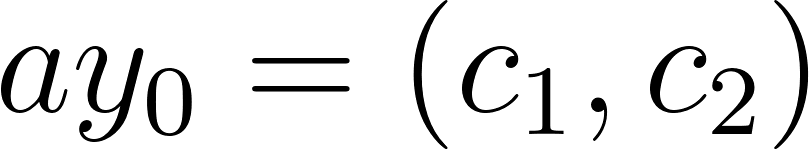
#### ElGamal

En ElGamal, la premisa es cifrar el mensaje a través de operaciones matemáticas basadas en la dificultad del problema del logaritmo discreto. Utilizamos un grupo cíclico de números primos y generamos claves que consisten en un elemento del grupo y su correspondiente exponente. El sistema criptográfico se define de la siguiente manera:

* El conjunto de texto plano son caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de texto cifrado es [](https://www.codecogs.com/eqnedit.php?latex=C%3DP%5Ctimes%20P#0).
* El conjunto de claves [](https://www.codecogs.com/eqnedit.php?latex=K%3D%5C%7B(p%2C%5Calpha%2Ca%2C%5Cbeta)%20%7C%20%5Cbeta%20%5Cequiv%20%5Calpha%5Ea%20%5Ctext%7Bmod%7D%20p%5C%7D#0)
* Para un número aleatorio secreto [](https://www.codecogs.com/eqnedit.php?latex=k#0), se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_%7BK%7D(x%2Ck)%3D(y_1%2Cy_2).%20%5C%2C%20y_1%3D%5Calpha%5Ek%20%5Ctext%7B%20mod%20%7D%20p%2C%5C%2C%20y_2%3Dx%5Cbeta%5Ek%20%5Ctext%7B%20mod%20%7D%20p.#0) También se define la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_K(y_1%2Cy_2)%3Dy_2(y%5Ea_1)%5E%7B-1%7D%20%5Ctext%7B%20mod%20%7Dp#0).

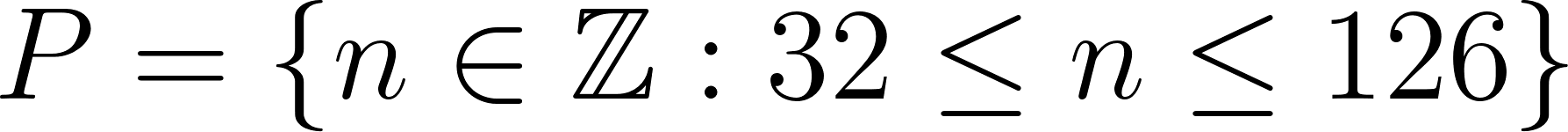
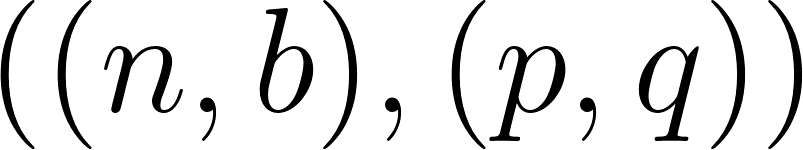
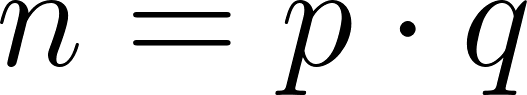
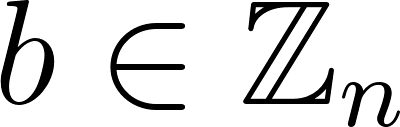
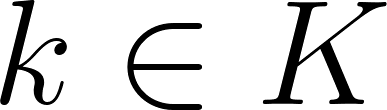
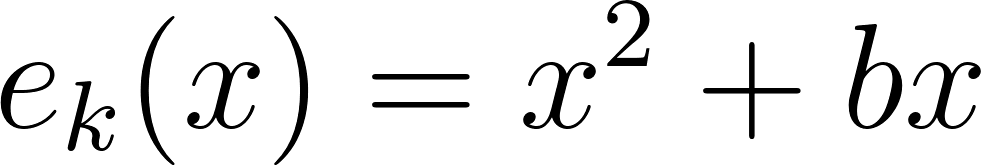
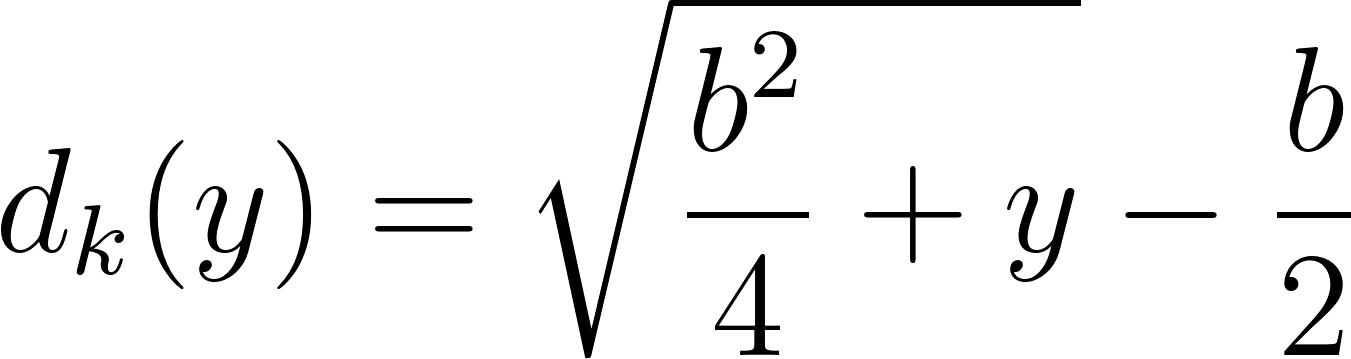
#### Menezes-Vanstone

En el sistema criptográfico de Menezes-Vanstone, el enfoque se centra en la operación de productos escalares en un espacio vectorial finito. La seguridad se basa en la dificultad del problema del logaritmo discreto en este contexto. La estructura del criptosistema se describe de la siguiente manera:

* Ahora definimos una curva elíptica sobre el espacio del texto plano con caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%3D%5Cmathbb%7BZ%7D%5E*%5Ctimes%20%5Cmathbb%7BZ%7D%5E*#0) y [](https://www.codecogs.com/eqnedit.php?latex=C%3DE%5Ctimes%20%5Cmathbb%7BZ%7D%5E*%5Ctimes%20%5Cmathbb%7BZ%7D%5E*#0).
* El conjunto de claves está dado por [](https://www.codecogs.com/eqnedit.php?latex=K%3D%5C%7B(E%2C%5Calpha%2Ca%2C%5Cbeta)%3A%5Cbeta%3Da%5Calpha%5C%7D#0), con [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%5Cin%20E#0), los valores de [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha%2C%20%5Cbeta#0) son públicos y [](https://www.codecogs.com/eqnedit.php?latex=a#0) es secreto.
* Para un número aleatorio secreto [](https://www.codecogs.com/eqnedit.php?latex=k#0) y para [](https://www.codecogs.com/eqnedit.php?latex=x%3D(x_1%2Cx_2)%5Cin%20%5Cmathbb%7BZ%7D%5E*%5Ctimes%20%5Cmathbb%7BZ%7D%5E*#0), se define la función de cifrado como [](https://www.codecogs.com/eqnedit.php?latex=e_K(x%2Ck)%3D(y_0%2Cy_1%2Cy_2)#0) con [](https://www.codecogs.com/eqnedit.php?latex=y_0%3Dk%5Calpha#0), [](https://www.codecogs.com/eqnedit.php?latex=(c_1%2Cc_2)%3Dk%5Cbeta#0), [](https://www.codecogs.com/eqnedit.php?latex=y_1%3Dc_1x_1%20%5Ctext%7B%20mod%20%7Dp#0) y [](https://www.codecogs.com/eqnedit.php?latex=y_2%3Dc_2x_2%20%5Ctext%7B%20mod%20%7Dp#0). También se define la función de descifrado como [](https://www.codecogs.com/eqnedit.php?latex=d_K(y)%3D(y_1c%5E%7B-1%7D_1%20%5Ctext%7B%20mod%20%7Dp%2C%20y_2c%5E%7B-1%7D_2%20%5Ctext%7B%20mod%20%7Dp)#0) y [](https://www.codecogs.com/eqnedit.php?latex=ay_0%3D(c_1%2Cc_2)#0).

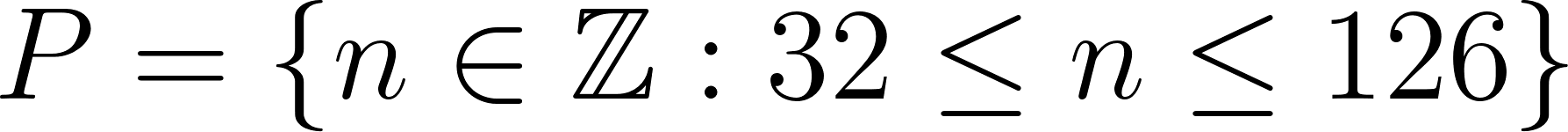
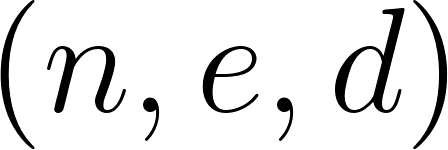
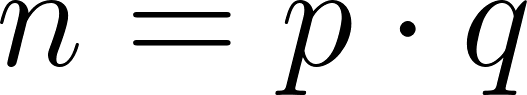
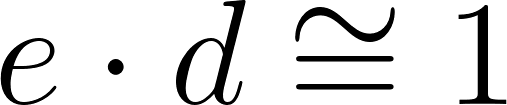
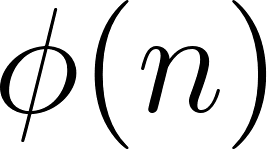
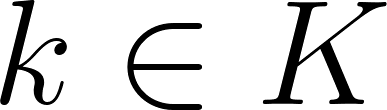
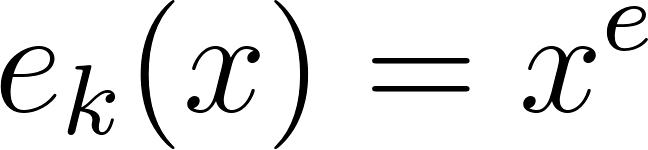
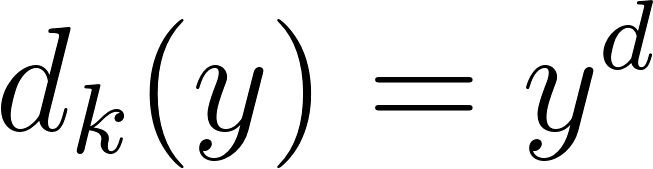
#### Rabin

En Rabin, buscamos cifrar el mensaje mediante una ecuación cuadrática, usando como base aritmética un par de primos lo suficientemente grandes como para que el problema de la factorización proteja nuestro mensaje de ataques externos. El criptosistema se define así:

* El conjunto de texto plano son caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de texto cifrado es [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbb%7BZ%7D%5E*#0)
* El conjunto de claves son duplas de la forma [](https://www.codecogs.com/eqnedit.php?latex=(%20(n%2Cb)%2C%20(p%2Cq)%20)#0). Donde [](https://www.codecogs.com/eqnedit.php?latex=n%3Dp%5Ccdot%20q#0) y [](https://www.codecogs.com/eqnedit.php?latex=b%5Cin%5Cmathbb%7BZ%7D_n#0)
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%3Dx%5E2%2Bbx#0) mod [](https://www.codecogs.com/eqnedit.php?latex=n#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%3D%5Csqrt%7B%5Cdfrac%7Bb%5E2%7D%7B4%7D%2By%7D-%5Cdfrac%7Bb%7D%7B2%7D#0) mod [](https://www.codecogs.com/eqnedit.php?latex=n#0)

#### RSA

En el cifrado RSA la persona que desea recibir el mensaje, digamos Alice, debe publicar su clave *(n,e) ￼*con la cual el mensajero, digamos Bob, deberá encriptar el texto plano para que así Alice lo desencripte con su clave privada. Este criptosistema está definido de la siguiente forma:

* El conjunto de texto plano son caracteres ASCII imprimibles [](https://www.codecogs.com/eqnedit.php?latex=P%20%3D%20%20%5C%7Bn%20%5Cin%20%5Cmathbb%7BZ%7D%20%3A%2032%20%5Cleq%20n%20%5Cleq%20126%20%5C%7D#0)
* El conjunto de texto cifrado es [](https://www.codecogs.com/eqnedit.php?latex=%5Cmathbb%7BZ%7D_n#0)
* El conjunto de claves son triplas de la forma [](https://www.codecogs.com/eqnedit.php?latex=(n%2Ce%2Cd)#0) donde las dos primeras componentes son la clave pública y la tercera la privada. Además, [](https://www.codecogs.com/eqnedit.php?latex=n%3Dp%5Ccdot%20q#0) donde [](https://www.codecogs.com/eqnedit.php?latex=p#0) y [](https://www.codecogs.com/eqnedit.php?latex=q#0) son primos lo suficientemente grandes; y [](https://www.codecogs.com/eqnedit.php?latex=e%5Ccdot%20d%5Ccong%201%20#0) mod [](https://www.codecogs.com/eqnedit.php?latex=%5Cphi(n)#0)
* Para [](https://www.codecogs.com/eqnedit.php?latex=k%20%5Cin%20K#0) se define la función de cifrado [](https://www.codecogs.com/eqnedit.php?latex=e_k(x)%20%3Dx%5Ee#0) mod [](https://www.codecogs.com/eqnedit.php?latex=n#0) y la función de descifrado [](https://www.codecogs.com/eqnedit.php?latex=d_k(y)%20%3D%20y%5Ed#0) mod [](https://www.codecogs.com/eqnedit.php?latex=n#0).

### Criptoanálisis para Vigenère

Este cifrado es una versión sofisticada del cifrado César. Se eligen múltiples claves de cifrado César y se aplican periódicamente.

Para descifrarlo, primero se debe encontrar la longitud de la clave. Se puede utilizar una técnica estadística (Índice de Coincidencia - IoC). En el texto en inglés, la probabilidad de que dos letras elegidas al azar sean iguales es diferente que en un texto aleatorio. Esto se debe a que, en inglés, las letras tienen probabilidades desiguales, siendo algunas más probables que otras. En un texto aleatorio, todas tienen la misma probabilidad. Conociendo la longitud de la clave, simplemente se tiene que romper muchos cifrados César.

## 

## Ejemplos de uso

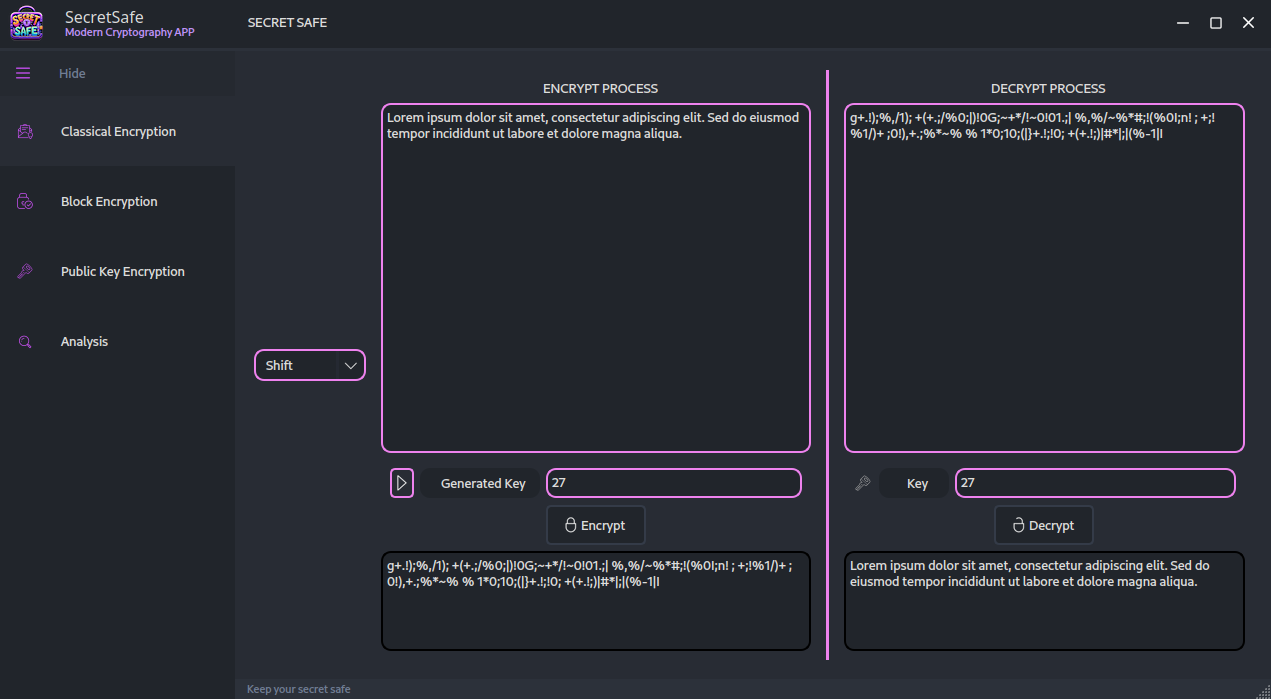
### Criptografía clásica

Para hacer uso de esta funcionalidad, se accede a la opción de “Classical Encryption” ubicada en el primer lugar del menú principal de la aplicación.

#### Desplazamiento

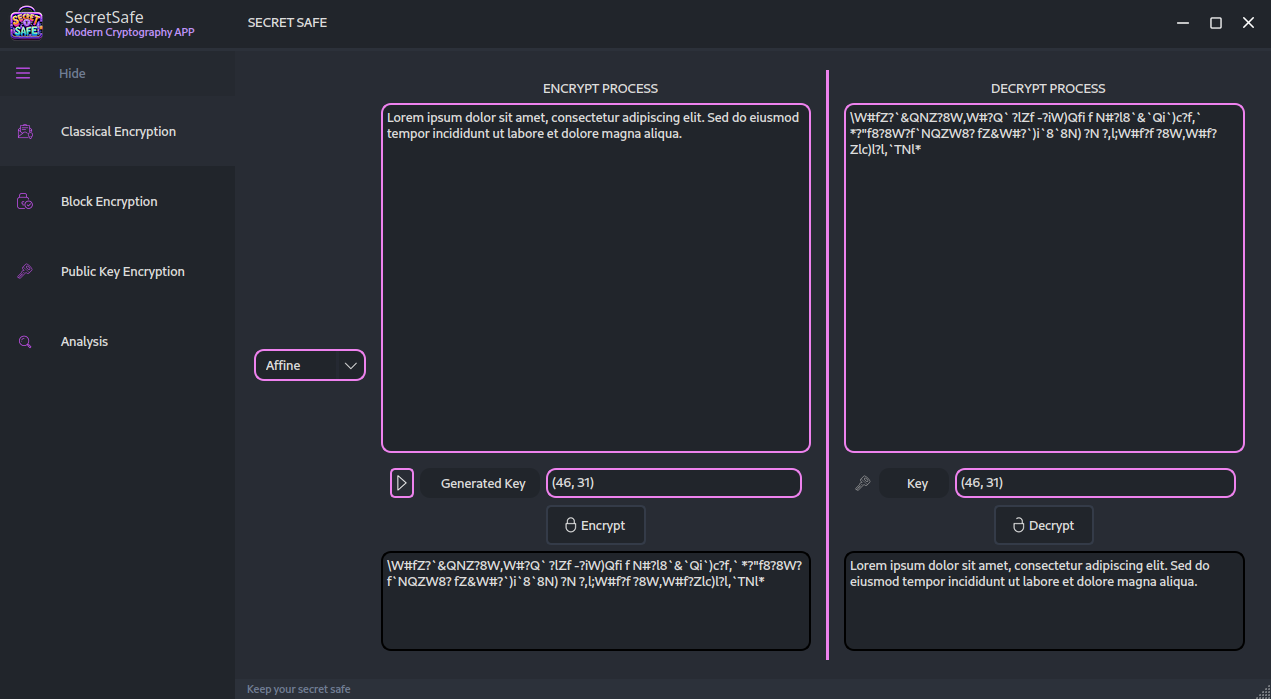
En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Shift” en el menú de la izquierda de la ventana de Criptografía Clásica. En la caja de “Encrypt Process” se ingresa el texto a encriptar. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en el cifrado por desplazamiento es un número entero. Finalmente, para encriptar el texto, se pulsa el botón de “Encrypt”.

El proceso de descifrado, se implementa de manera similar. El texto cifrado se ingresa en la caja de texto de “Decrypt Process”. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja de texto bajo este botón.



#### 

#### Afín

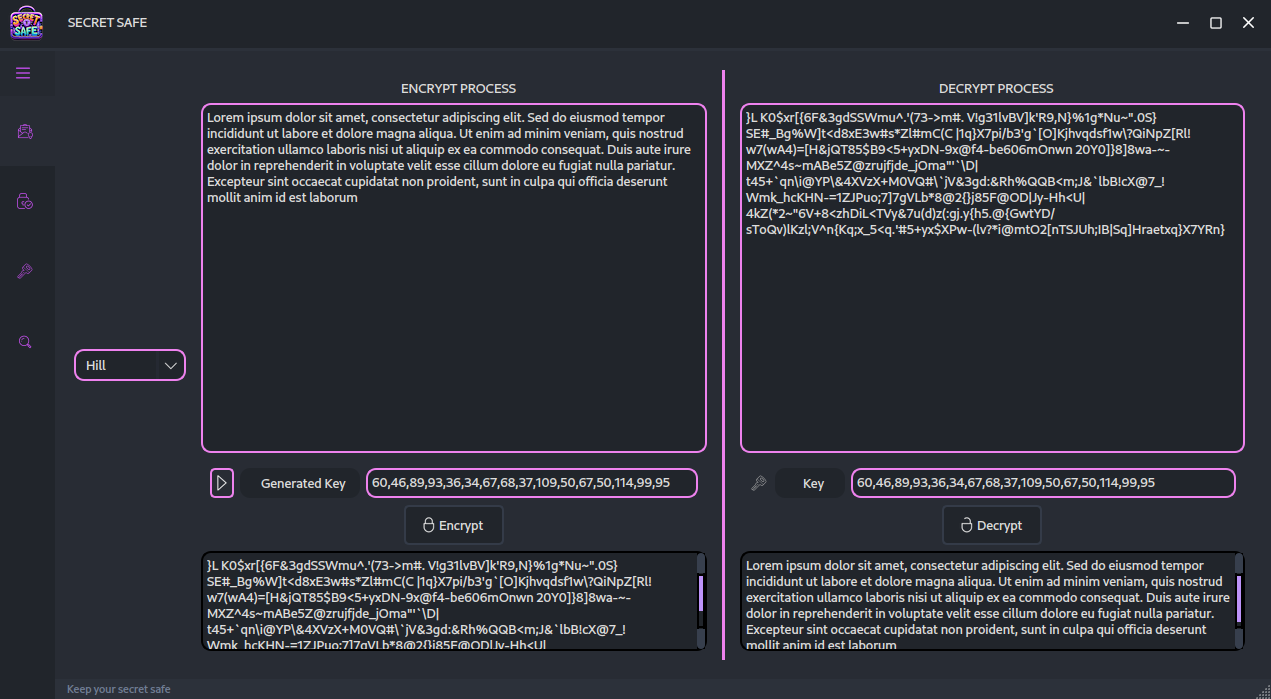
En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Affine” en el menú de la izquierda de la ventana de Criptografía Clásica. En la caja de “Encrypt Process” se ingresa el texto a encriptar. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en el cifrado afin es una pareja de enteros. Finalmente, para encriptar el texto, se pulsa el botón de “Encrypt”.  
El proceso de descifrado, se implementa de manera similar. El texto cifrado se ingresa en la caja de texto de “Decrypt Process”. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja de texto bajo este botón.  


#### 

#### Hill

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Hill” en el menú de la izquierda de la ventana de Criptografía Clásica. En la caja de “Encrypt Process” se ingresa el texto a encriptar. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en el cifrado Hill es una matriz de 4*×*4 la cual se entrega como una línea de 16 números separados por coma. Finalmente, para encriptar el texto, se pulsa el botón de “Encrypt”.

El proceso de descifrado, se implementa de manera similar. El texto cifrado se ingresa en la caja de texto de “Decrypt Process”. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja de texto bajo este botón.

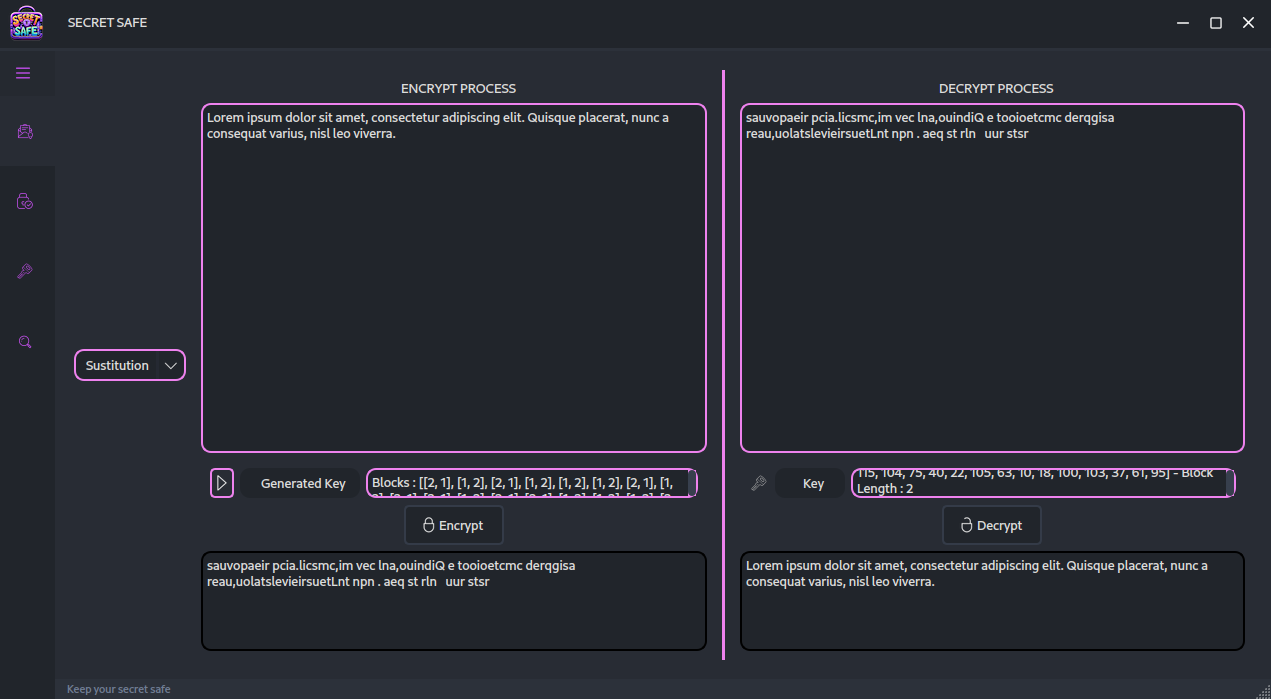


#### 

#### Sustitución-Permutación

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Sustitution” en el menú de la izquierda de la ventana de Criptografía Clásica. En la caja de “Encrypt Process” se ingresa el texto a encriptar. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado consiste en dos partes. La primera son permutaciones en bloque de menor tamaño al texto plano. La segunda parte es una permutación del tamaño total del texto. Además, se entrega el tamaño del bloque para las permutaciones pequeñas. Finalmente, para encriptar el texto, se pulsa el botón de “Encrypt”.

El proceso de descifrado, se implementa de manera similar. El texto cifrado se ingresa en la caja de texto de “Decrypt Process”. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja de texto bajo este botón.

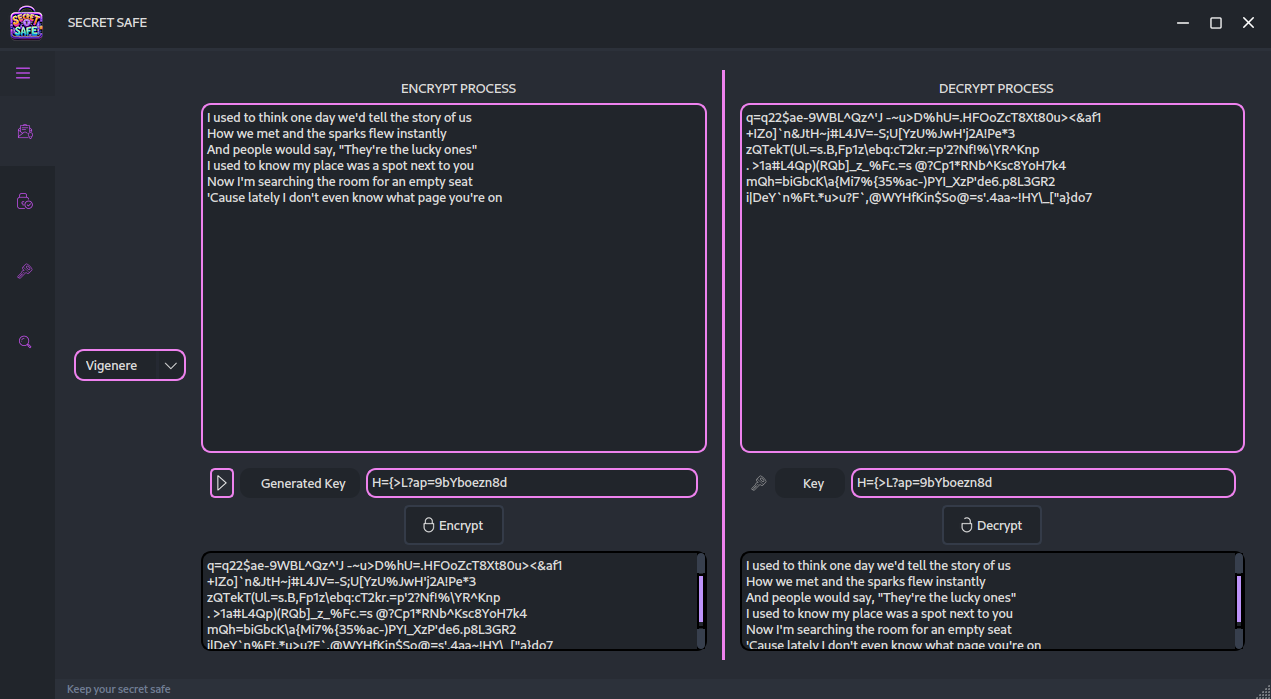


#### 

#### Vigenère

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Vigenère” en el menú de la izquierda de la ventana de Criptografía Clásica. En la caja de “Encrypt Process” se ingresa el texto a encriptar. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado es una cadena aleatoria de la mitad de tamaño que el texto a cifrar. Esta cadena también pertenece al conjunto de texto plano. Finalmente, para encriptar el texto, se pulsa el botón de “Encrypt”.

El proceso de descifrado, se implementa de manera similar. El texto cifrado se ingresa en la caja de texto de “Decrypt Process”. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja de texto bajo este botón.



### 

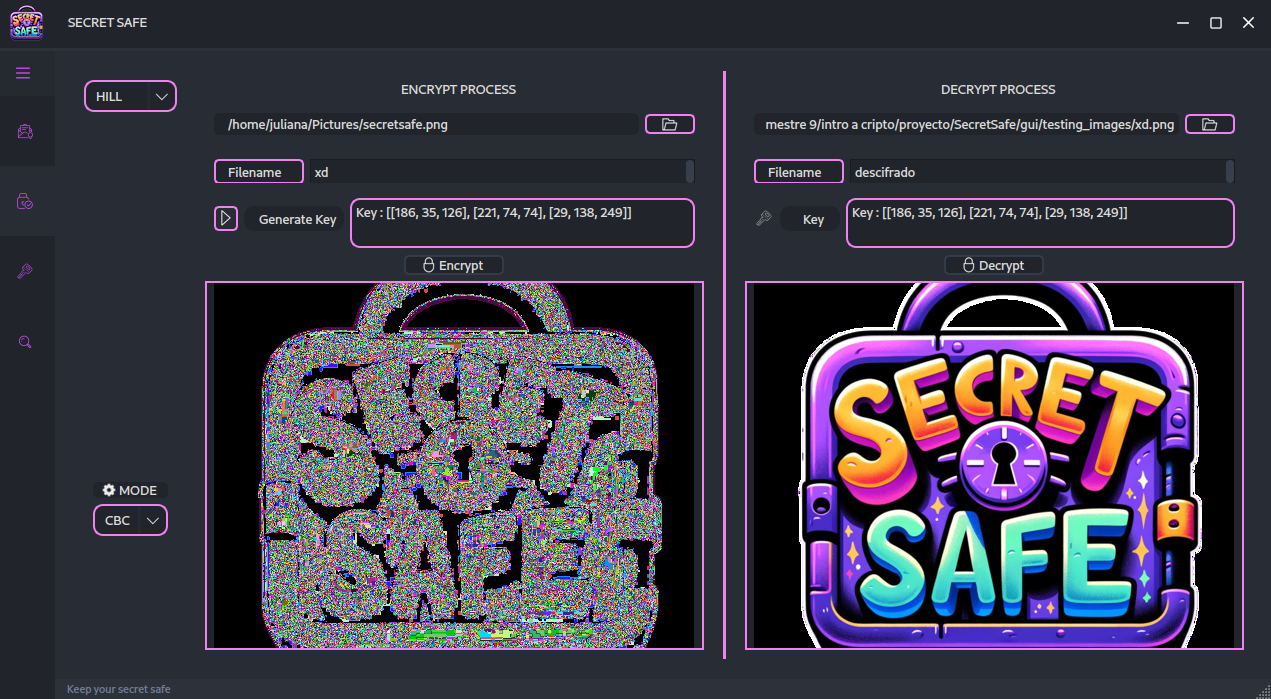
### Criptografía de imágenes

Para hacer uso de esta funcionalidad, se accede a la opción de “Image Encryption” ubicada en el segundo lugar del menú principal de la aplicación.

#### Hill

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “Hill” en el menú de la parte superior izquierda de la ventana de “Image Encryption”. En la caja de “Encrypt Process” se ingresa la imagen a encriptar. Después, se ingresa el nombre con el que se quiere guardar la imagen. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado es una matriz 3x3 invertible. El encriptado de imagen se realiza usando un modo de cifrado en bloque. Este modo se puede elegir en el menú de la parte izquierda de la ventana de “Image Encryption”. Finalmente, para encriptar la imagen, se pulsa el botón de “Encrypt”. La imagen cifrada se aloja en la carpeta del repositorio local gui/tested\_images.

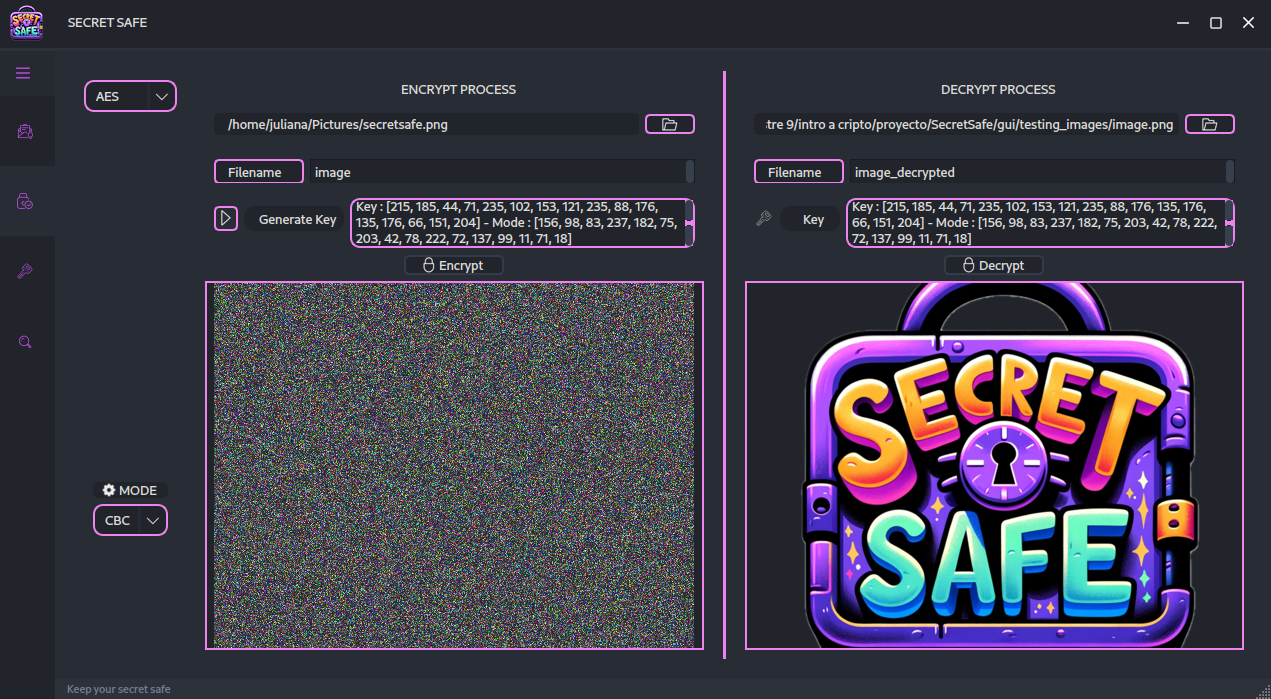
El proceso de descifrado, se implementa de manera similar. La imagen cifrada se ingresa en “Decrypt Process”. Se ingresa el nombre de la imagen descifrada. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja bajo este botón.



#### AES

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “AES” en el menú de la parte superior izquierda de la ventana de “Image Encryption”. En la caja de “Encrypt Process” se ingresa la imagen a encriptar. Después, se ingresa el nombre con el que se quiere guardar la imagen. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado consiste de 16 bits, también se proporciona la información necesaria para ejecutar el modo seleccionado ya sea para cifrado o descifrado . El encriptado de imagen se realiza usando un modo de cifrado en bloque. Este modo se puede elegir en el menú de la parte izquierda de la ventana de “Image Encryption”. Finalmente, para encriptar la imagen, se pulsa el botón de “Encrypt”. La imagen cifrada se aloja en la carpeta del repositorio local gui/tested\_images.

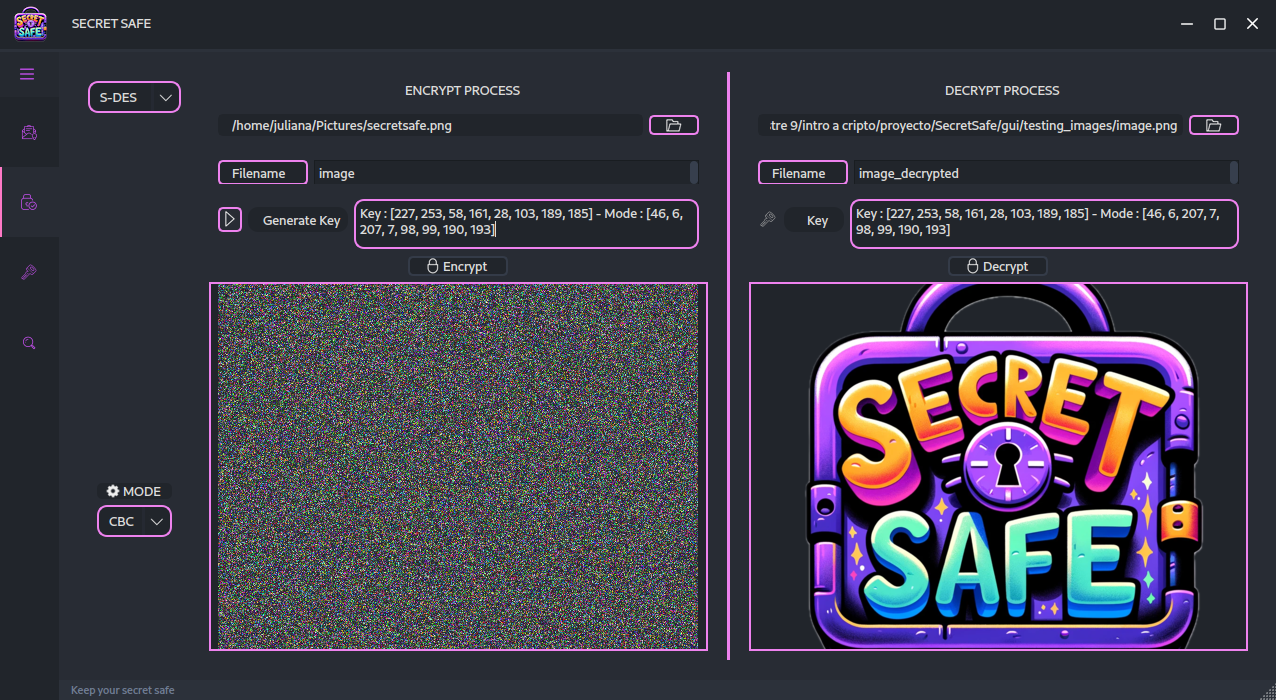
El proceso de descifrado, se implementa de manera similar. La imagen cifrada se ingresa en “Decrypt Process”. Se ingresa el nombre de la imagen descifrada. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado

se mostrará en la caja bajo este botón.  


#### SDES

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “SDES” en el menú de la parte superior izquierda de la ventana de “Image Encryption”. En la caja de “Encrypt Process” se ingresa la imagen a encriptar. Después, se ingresa el nombre con el que se quiere guardar la imagen. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado consiste de 8 bits, también se proporciona la información necesaria para ejecutar el modo seleccionado ya sea para cifrado o descifrado . El encriptado de imagen se realiza usando un modo de cifrado en bloque. Este modo se puede elegir en el menú de la parte izquierda de la ventana de “Image Encryption”. Finalmente, para encriptar la imagen, se pulsa el botón de “Encrypt”. La imagen cifrada se aloja en la carpeta del repositorio local gui/tested\_images.

El proceso de descifrado, se implementa de manera similar. La imagen cifrada se ingresa en “Decrypt Process”. Se ingresa el nombre de la imagen descifrada. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja bajo ese botón.

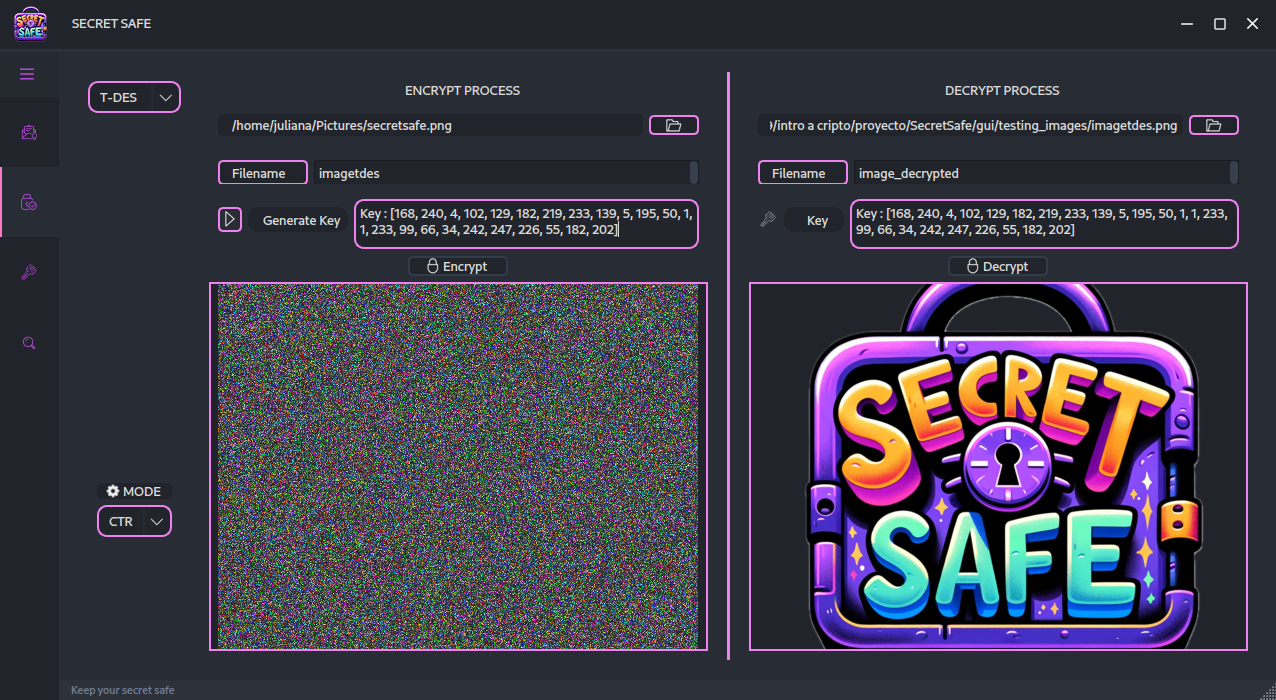


#### 

#### Triple-DES

En SecretSafe, este tipo de cifrado se puede usar seleccionando la opción “TDES” en el menú de la parte superior izquierda de la ventana de “Image Encryption”. En la caja de “Encrypt Process” se ingresa la imagen a encriptar. Después, se ingresa el nombre con el que se quiere guardar la imagen. Luego, se genera la clave pulsando el botón ubicado al lado izquierdo de “Generated Key”. La clave en este cifrado consiste de 8 bits, también se proporciona la información necesaria para ejecutar el modo seleccionado ya sea para cifrado o descifrado . El encriptado de imagen se realiza usando un modo de cifrado en bloque. Este modo se puede elegir en el menú de la parte izquierda de la ventana de “Image Encryption”. Finalmente, para encriptar la imagen, se pulsa el botón de “Encrypt”. La imagen cifrada se aloja en la carpeta del repositorio local gui/tested\_images.

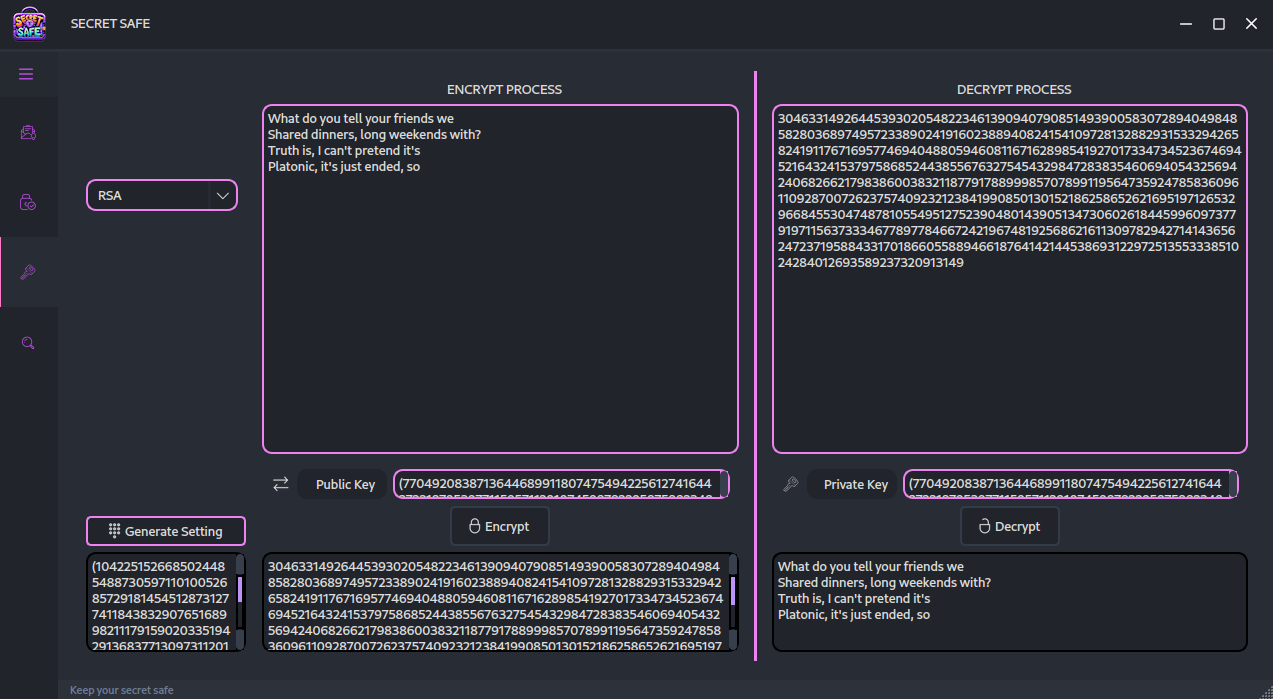
El proceso de descifrado, se implementa de manera similar. La imagen cifrada se ingresa en “Decrypt Process”. Se ingresa el nombre de la imagen descifrada. La clave para descifrar se ingresa en el campo de texto de “Key”. Finalmente, se pulsa el botón de “Decrypt” y el resultado se mostrará en la caja bajo ese botón.



### 

### Criptografía de clave pública

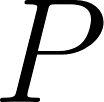
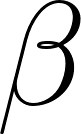
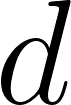
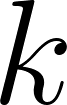
#### ElGamal

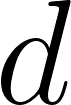
En SecretSafe, encontramos este cifrado en el apartado de llave pública, eligiendo la opción “Gamal” en la lista desplegable de la parte izquierda. Este cifrado puede cifrar mensajes de texto con un máximo de 200 caracteres de longitud. Para hacer esto, se ingresa el texto en la caja de “Encrypt Process”. Luego se genera el par de llave pública y privada pulsando en la opción de “Generate Setting” en la parte inferior izquierda de la ventana, esto puede tardar algunos segundos. Lo anterior mostrará la llave pública y privada en sus respectivos campos de texto. Para cifrar el texto se oprime el botón “Encrypt”. El proceso de descifrado es similar. Es importante garantizar que la llave privada este en el campo correspondiente para realizar este proceso  


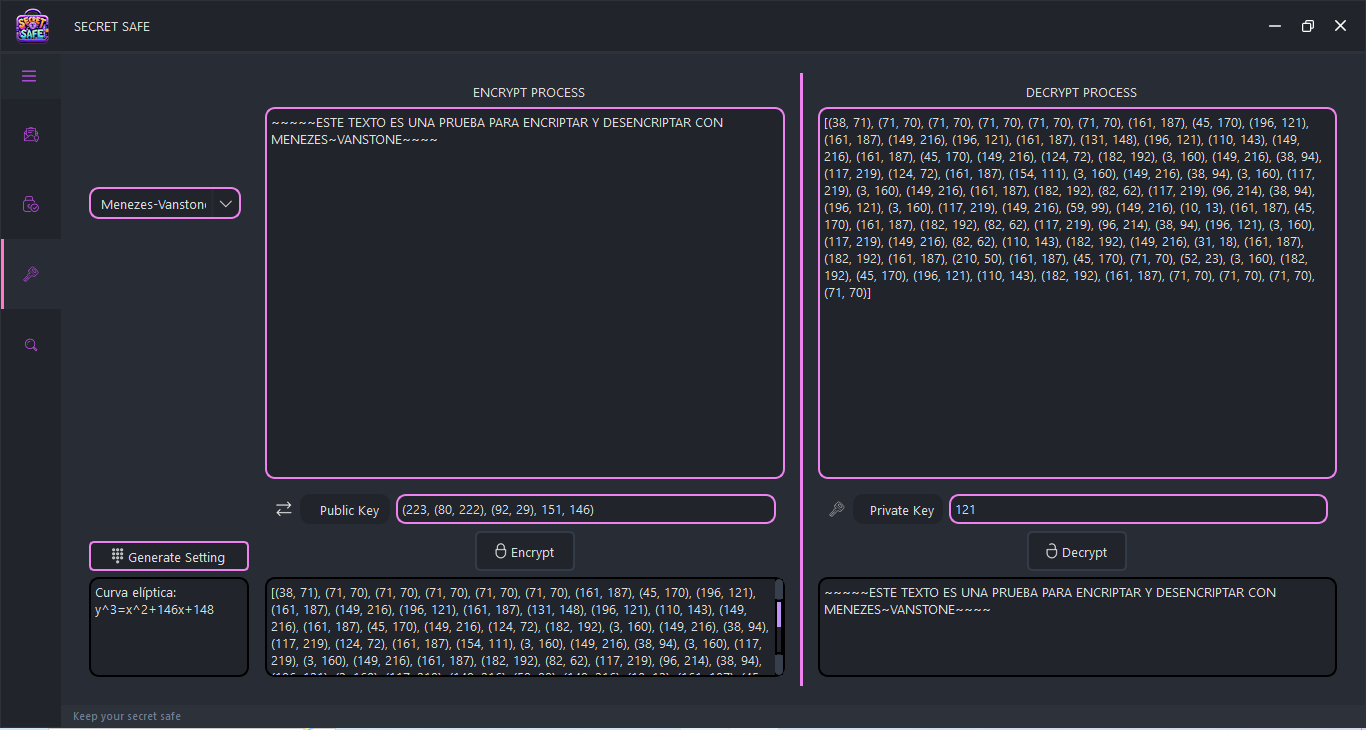
#### Menezes-Vanstone

Para encriptar o desencriptar con este tipo de cifrado en SecretSafe, primero elegimos el apartado de llave pública, luego en el menú desplegable se debe elegir “Menezes-Vanstone”. Aquí el usuario ingresa un texto plano, y al oprimir el botón “Generate Setting”, la aplicación genera las claves públicas y privadas necesarias, adicionalmente le mostrará al usuario la curva elíptica generada para construir el sistema criptográfico.

En el cuadro de clave pública aparece una quíntupla:

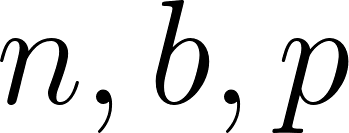
* [](https://www.codecogs.com/eqnedit.php?latex=P#0) es el primer valor, el primo que rige la aritmética del sistema.
* [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0) el punto generador en nuestro grupo de puntos de la curva elíptica.
* [](https://www.codecogs.com/eqnedit.php?latex=%5Cbeta#0) es [](https://www.codecogs.com/eqnedit.php?latex=d#0) veces el punto [](https://www.codecogs.com/eqnedit.php?latex=%5Calpha#0).
* [](https://www.codecogs.com/eqnedit.php?latex=k#0) un entero que nos garantiza la no tratabilidad del problema del logaritmo.
* [](https://www.codecogs.com/eqnedit.php?latex=a#0) el factor que acompaña a [](https://www.codecogs.com/eqnedit.php?latex=x#0) en la curva elíptica.

Por otra parte, la clave privada es simplemente el entero [](https://www.codecogs.com/eqnedit.php?latex=d#0).

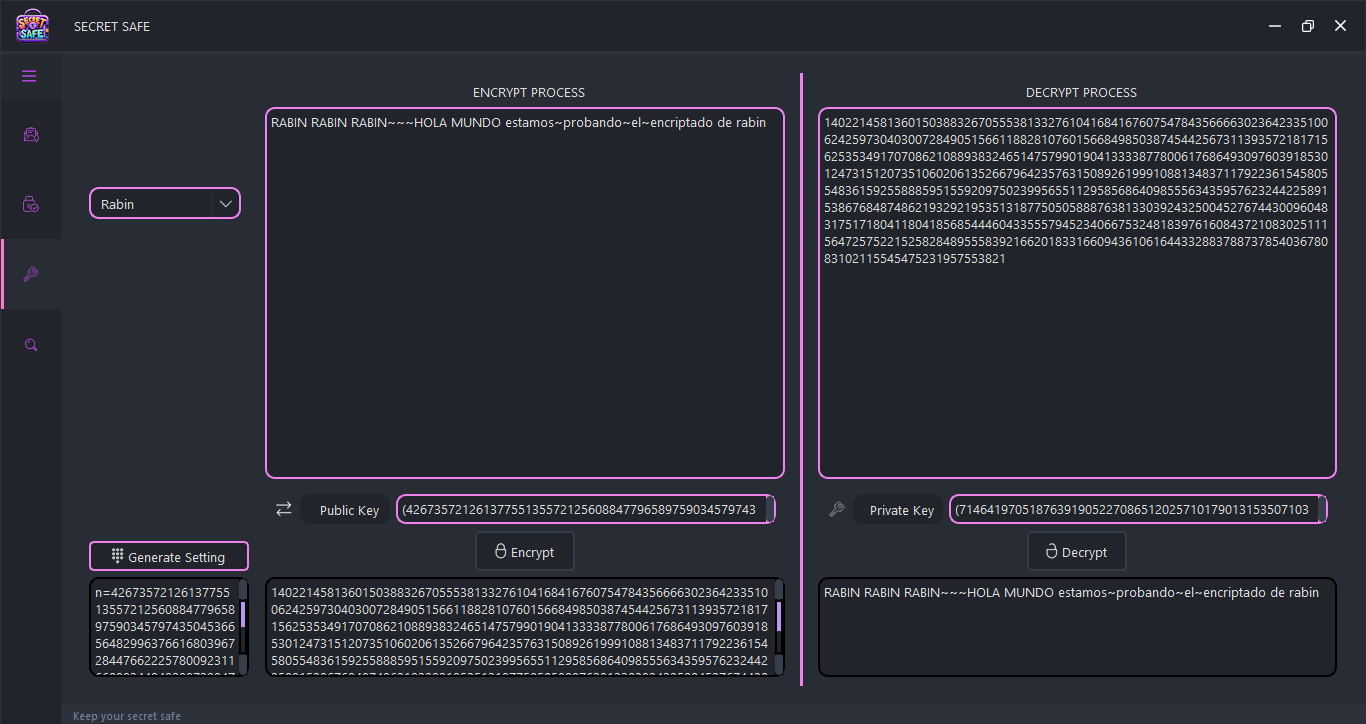
Así, el usuario puede encriptar su texto conociendo todos los valores necesarios para generar su texto encriptado. De manera análoga, puede desencriptar un mensaje cifrado sin mayor problema.

#### 

#### Rabin

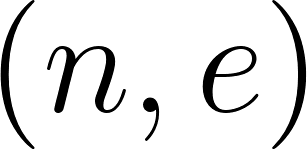
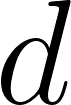
En SecretSafe encontramos este cifrado en el apartado de llave pública, eligiendo “Rabin” en la lista desplegable, la segunda opción. Primero, se puede generar la clave con “Generate setting”, aquí SecretSafe va a crear las claves públicas y privadas, las anexa a los campos correspondientes, luego el usuario ingresa su texto en ASCII, y oprime el botón “Encrypt”. En el cuadro de texto de configuración se mostrarán las 2 componentes de cada clave, es decir [](https://www.codecogs.com/eqnedit.php?latex=n%2Cb%2Cp#0) y [](https://www.codecogs.com/eqnedit.php?latex=q#0).

Para desencriptar, podemos copiar y pegar el texto deseado; si es con el mismo sistema generado anteriormente dejamos la misma clave privada y nos retorna el texto original al oprimir “Decrypt” como vemos a continuación:



#### 

#### RSA

En SecretSafe encontramos este cifrado en el apartado de llave pública, eligiendo “RSA” en la lista desplegable. Aquí el usuario ingresa un texto plano, y al oprimir el botón “Generate Setting”, la aplicación genera las claves públicas y privadas necesarias. En este caso mostrará la pareja [](https://www.codecogs.com/eqnedit.php?latex=(n%2Ce)#0) como clave pública y el número [](https://www.codecogs.com/eqnedit.php?latex=d#0) como clave privada. Adicionalmente, en el cuadro de texto de configuración mostrará los primos [](https://www.codecogs.com/eqnedit.php?latex=p#0) y [](https://www.codecogs.com/eqnedit.php?latex=q#0) con los que generamos [](https://www.codecogs.com/eqnedit.php?latex=n#0).

Al encriptar, dejará el mensaje cifrado en forma de un número entero.

Para desencriptar, es el mismo proceso, pero oprimiendo el botón “Decrypt”, lo que nos retorna el mensaje original como en el siguiente ejemplo:

#### 

### 

### Criptoanálisis para Vigenère

En SecretSafe el criptoanalisis de Vigenère se realiza con mensajes de texto con caracteres en el alfabeto en inglés (ASCII 65 - 90). El mensaje debe ser lo suficientemente largo para que cada letra del alfabeto aparezca al menos una vez. Para hacer el análisis, se ingresa el mensaje de texto en la caja de “Encrypt Process”. Y se pulsa el botón de “Analyze”. En la caja al lado derecho de la pantalla se mostrará el análisis realizado que puede contener varias opciones con sus respectivos posibles tamaños para la clave.  
