Protegiendo Mensajes en una Red Segura con Cifrado Vigenère

En una empresa de telecomunicaciones, los empleados necesitan enviar mensajes cifrados para evitar que información confidencial sea interceptada por terceros. Sin embargo, debido a la sensibilidad de la información, las claves utilizadas para el cifrado son extremadamente largas, tanto que no pueden almacenarse en una sola variable convencional.

Para resolver este problema, usted debe implementar un sistema de cifrado basado en Vigenère, en el cual la clave será tratada como un número de gran tamaño almacenado en un arreglo de tamaño fijo, y el alfabeto estará representado en una matriz bidimensional.

Objetivo del Laboratorio

Implementar una clase `BigVigenere` en **Java** que permite cifrar y descifrar mensajes utilizando el cifrado de Vigenère con una clave numérica extremadamente larga, almacenada en un arreglo de tamaño fijo, y un alfabeto representado en una matriz.

Especificación de la clase `BigVigenere`

Debe contar con los siguientes atributos:

- `int[] key`: Arreglo de enteros que almacena la clave numérica de gran tamaño.
- `char[][] alphabet`: Matriz bidimensional que representa el alfabeto utilizado en el cifrado, compuesto por caracteres alfanuméricos (`'a' - 'z'`, `‘A - Z’` y `'0' - '9'`).

Y los siguientes métodos:

- `public BigVigenere()`: Constructor que inicializa la `key` como vacía. Luego le solicita una al usuario para hacer un set del atributo. Y también genera la matriz del alfabeto para el algoritmo Vigenère.
- `public BigVigenere(String numericKey)`: Constructor que recibe la clave numérica como una cadena y la almacena en el arreglo `key`. Y también genera la matriz del alfabeto para el algoritmo Vigenère.
- `public String encrypt(String message)`: Método que cifra un mensaje utilizando la clave numérica almacenada y la matriz del alfabeto usando el algoritmo de Vigenère.
- `public String decrypt(String encryptedMessage)`: Método que descifra un mensaje cifrado con la misma clave numérica usando Vigenère.

- `public void reEncrypt()`: Método que permite cambiar la clave utilizada en el cifrado. Con el siguiente paso a paso:
    - Se debe solicitar el mensaje encriptado.
    - Se debe descifrar el mensaje actual con la clave actual (como input desde la terminal).
    - A continuación se debe solicitar la nueva clave.
    - Luego se debe cifrar con la nueva clave.
    - Por último se imprime el nuevo mensaje.

- `public char search(int position):` Método que busca la letra correspondiente a la posición buscada, se realiza una búsqueda iterativa de izquierda a derecha, arriba hacia abajo. Retorna el carácter de la posición buscada.
- `public char optimalSearch(int position):` Método que realiza la búsqueda del carácter correspondiente de acuerdo a la posición indicada. Se busca realizar una búsqueda más eficiente que el caso anterior.

Experimentos:

1. **Diseñar y ejecutar pruebas unitarias** para verificar que los métodos `encrypt` y `decrypt` funcionen correctamente.
2. **Implementar la clase `VigenereCipher`.**

3. **Evaluar y gráficar el tiempo de ejecución** del cifrado y descifrado con claves de distintos tamaños `L = {10, 50, 100, 500, 1000, 5000}` y un mismo mensaje (El mensaje a cifrar debe tener al menos 10000 carácteres).

4. **Comparar la eficiencia del cifrado** a medida que aumenta el tamaño de la clave analizar el tiempo de ejecucion con la notación Big O.

¡¡Mucho éxito!! Si tiene dudas, PREGUNTE.