Implementación de Criptografía de Clave Pública y Privada en Java

La criptografía de clave pública y clave privada es un método seguro de cifrado que utiliza un par de claves relacionadas matemáticamente: una clave pública y una clave privada. Estas claves garantizan que lo que se cifra con una clave sólo pueda descifrarse con la otra, lo que asegura la confidencialidad y autenticidad de los mensajes en línea.

Cada usuario que participa en la comunicación tiene su propio par de claves: una clave pública que comparte con otros y una clave privada que mantiene en secreto. La clave pública se utiliza para encriptar los mensajes, mientras que la clave privada se utiliza para desencriptarlos. Esto permite que cualquiera pueda enviar un mensaje cifrado a un destinatario utilizando su clave pública, pero sólo el destinatario posee la clave privada necesaria para descifrarlo.

En este programa, se simula la comunicación entre dos personas, Alicia y Álvaro, utilizando la clase Encriptador. Cada una de ellas instancia un objeto Encriptador, lo que les proporciona su propio par de claves. Para enviar un mensaje seguro, Alicia utiliza la clave pública de Álvaro para cifrar el mensaje antes de enviarlo. Álvaro, al recibir el mensaje cifrado, utiliza su clave privada para descifrarlo y leer el mensaje original. Del mismo modo, Álvaro puede responder de manera segura a Alicia utilizando el mismo proceso.

El programa está estructurado de la siguiente manera:

Clase Encriptador:
Esta clase proporciona métodos para encriptar y desencriptar mensajes utilizando el algoritmo de encriptación RSA (Rivest-Shamir-Adleman), que es uno de los más utilizados. Dentro de la clase Encriptador encontraremos:

- **Atributos:**
  privateKey: Representa la clave privada utilizada para desencriptar mensajes.
  publicKey: Representa la clave pública utilizada para encriptar mensajes.
- **Constructor:** En el constructor se inicializan las claves pública y privada mediante la clase KeyPairGenerator, utilizando el método getInstance() y pasándole como argumento el algoritmo de encriptación que vamos a usar, RSA. Con el método initialize() vamos a especificar que el tamaño de las claves será de 2048 bits. Con el método generateKeyPair() generamos el par de claves.
- **Métodos:**
  -getPublicKey(): Devuelve la clave pública.
  -encriptar(String mensaje, PublicKey publicKey): Este método toma un mensaje de texto sin formato y la clave pública de destino, y devuelve el mensaje encriptado como una cadena de texto en formato Base64. Para ello, crea una instancia de la clase Cipher usando el método getInstance() pasándole como argumento el algoritmo de encriptación. Con el método init de la clase Cipher le pasamos el modo de cifrado, en este caso encriptar y la clave pública del destinatario (Cipher.DECRYPT_MODE, privateKey). Por último usa el método doFinal() de la

clase Cipher para realizar la operación de cifrado y la clase Base64 para codificar los bytes del mensaje cifrado.
-desencriptar(String mensajeEncriptado): Este método toma un mensaje encriptado en formato Base64 y utiliza la clave privada almacenada en el objeto Encriptador para desencriptarlo. Devuelve el mensaje original. Es el mismo método que encriptar, sólo cambia el modo de encriptación que le pasamos al método init() y la clave que le pasamos.

- **Main**
  En el método principal se demuestra el funcionamiento de Encriptador. Para ello se crean dos objetos Encriptador que simulan dos usuarios, Alicia y Álvaro.
  Se encripta un mensaje que envía Alicia para que Álvaro lo desencripte y Álvaro responde de la misma manera para que Alicia lo desencripte.