Universidad de los Andes
Ingeniería de Sistemas y Computación
ISIS 2203 Infraestructura Computacional

Taller Librería de Seguridad

El propósito es estudiar algunos métodos de cifrado de información implementados en librerías Java (javax.crypto.*) . Este taller usa métodos particulares, pero estos métodos no son los únicos disponibles para cifrar y descifrar información.

Cifrado Simétrico:

Escriba una clase para ejecutar un algoritmo de cifrado simétrico. A continuación encuentra parte del código necesario para cifrar y descifrar información:

1. Definición de atributos de la clase:

```
private SecretKey desKey;
private final static String ALGORITMO="AES";
private final static String PADDING="AES/ECB/PKCS5Padding";
```

2. Método para cifrar

```
public byte[] cifrar() {
	byte [] cipheredText;

	try {
		KeyGenerator keygen = KeyGenerator.getInstance(ALGORITMO);
		desKey = keygen.generateKey();
		Cipher cipher = Cipher.getInstance(PADDING);

		BufferedReader stdIn = new BufferedReader(
				new InputStreamReader(System.in));
		String pwd = stdIn.readLine();
		byte [] clearText = pwd.getBytes();
		String s1 = new String (clearText);
		System.out.println("clave original: " + s1);

		cipher.init(Cipher.ENCRYPT_MODE, desKey);
		long startTime = System.nanoTime();
		cipheredText = cipher.doFinal(clearText);
		long endTime = System.nanoTime();
		String s2 = new String (cipheredText);
		System.out.println("clave cifrada: " + s2);
		System.out.println("Tiempo: " + (endTime - startTime));
		return cipheredText;
	}
	catch (Exception e) {
		System.out.println("Excepcion: " + e.getMessage());
		return null;
	}
}
```

3. Método para descifrar

```
public void descifrar(byte [] cipheredText) {

	try {
		Cipher cipher = Cipher.getInstance(PADDING);
		cipher.init(Cipher.DECRYPT_MODE, desKey);
		byte [] clearText = cipher.doFinal(cipheredText);
		String s3 = new String(clearText);
		System.out.println("clave original: " + s3);
	}
	catch (Exception e) {
		System.out.println("Excepcion: " + e.getMessage());
	}
}
```

Escriba una clase para crear una instancia de la clase que cifra y descifra información de forma simétrica.

1. En el main
   a. Cree una instancia de la clase que cifra y descifra información de forma simétrica
   b. Ejecute un llamado al método cifrar (con los parámetros apropiados)
   c. Ejecute un llamado al método descifrar (con los parámetros apropiados)
   d. Ejecute el programa y observe los resultados (el texto original y el texto descifrado deberían coincidir)
2. Responda las siguientes preguntas:
   a. ¿Qué significa ECB como modo de ejecución del algoritmo de cifrado?
   b. ¿Cuál es la ventaja de ECB comparado con CBC?
   c. ¿Qué es y para qué se necesita el padding?
3. Modifique el main
   a. Cree una segunda instancia de la clase que cifra y descifra información de forma simétrica
   b. Ejecute el programa de tal forma que ambas instancias cifren la misma cadena de entrada y observe los resultados
   c. ¿Qué pasa si cifra con la primera instancia y descifra el resultado con la segunda instancia?
4. Diseñe e implemente una clase nueva que permita
   a. Cifrar una cadena de texto de forma simétrica y almacenarla en un archivo
   b. Recuperar la información del archivo
   c. Observe que primero debe determinar qué información necesita para recuperar la cadena original y definir cómo manejar dicha información (cómo almacenarla y recuperarla)

Las instrucciones siguientes pueden ser usadas (también puede buscar otros métodos para almacenar y recuperar la información).

```
// pCifrado: texto cifrado
// pKey: llave
FileOutputStream farch = new FileOutputStream("datoCifrado");
ObjectOutputStream oos = new ObjectOutputStream(farch);
oos.writeObject(pCifrado);
oos.close();

FileOutputStream farch2 = new FileOutputStream("llave");
ObjectOutputStream oos2 = new ObjectOutputStream(farch2);
oos2.writeObject(pKey);
oos2.close();
```

```
// Recuperando texto cifrado
FileInputStream input = new FileInputStream("datoCifrado");
ObjectInputStream ois = new ObjectInputStream(input);
byte cipheredText[] = (byte[])ois.readObject();
ois.close();

FileInputStream input2 = new FileInputStream("llave");
ObjectInputStream ois2 = new ObjectInputStream(input2);
SecretKey llave = (SecretKey)ois2.readObject();
ois2.close();

// Construir cipher para descifrar
...
```