有關 Java 密碼程序

listing of gen_AES_key.java:

```
import java.io.*;
import java.security.*;

import javax.crypto.*;
import sun.misc.*;
import com.sun.crypto.provider.SunJCE;
import javax.crypto.spec.*;

public class gen_AES_key {
public static void main(String args[]) {
   try {
      System.out.println("\nCreating a AES key...");
      KeyGenerator kg = KeyGenerator.getInstance("AES");

      // print the Provider
      System.out.println("Provider: " + kg.getProvider().toString());

      // key size: 256 bits
      kg.init(256);
      SecretKey sessionKey = kg.generateKey();

      // Instantiate the cipher
      Cipher cipher = Cipher.getInstance("AES/OFB/PKCS5Padding");
      cipher.init(Cipher.ENCRYPT_MODE, sessionKey);

      // obtain the Initial Vector(IV)
      byte[] iv = cipher.getIV();
      // The OFB and CFB use an initialization vextor(IV) for operation and,
      // when these modes are used, the IV needs to be maintained.

      // create a file contain encrypt data
      ObjectOutputStream obj_out = new ObjectOutputStream(
      new CipherOutputStream(
      new BufferedOutputStream(
      new FileOutputStream("secret_objects.dat")), cipher));

      // writing a few simple object, eg String
      System.out.println("Writing secret data into file named \"secret_objects.dat\"...");
      obj_out.writeObject("AES demo...");
```

```
      obj_out.writeObject("Another AES demo...");

      obj_out.close();

      // Decrypt the objects
      cipher = Cipher.getInstance("AES/OFB/PKCS5Padding");
      IvParameterSpec spec = new IvParameterSpec(iv);
      cipher.init(Cipher.DECRYPT_MODE, sessionKey, spec);

      ObjectInputStream obj_in = new ObjectInputStream(
      new CipherInputStream(
      new BufferedInputStream(
      new FileInputStream("secret_objects.dat")), cipher));

      // reading back the objects from the file
      System.out.println("\nreading back the objects from the file...");
      System.out.println((String)obj_in.readObject());
      System.out.println((String)obj_in.readObject());
      obj_in.close();

   } catch (Exception e) {
      System.out.println(e.toString());
   }
}
}
```