

```

// Network Security - Assignment # 2
// by- Harsh & Nishtha
// Key generation for RSA

import java.io.*;
import java.security.spec.*;
import java.security.*;
import javax.crypto.*;
import javax.crypto.spec.*;

public class keygen
{
    KeyPair kp1, kp2;
    PrivateKey priv1, priv2;
    PublicKey publ, pub2;
    String KeyFile1="key.pair";
    String KeyFile2="key.pub";
    String KeyFile3="key.pri";

    public void setKeyName(String filename)
    {
        KeyFile1 = filename + ".pair";
        KeyFile2 = filename + ".pub";
        KeyFile3 = filename + ".pri";
    }

    public void KeyGeneration()
    {
        try{

            //KeyPairGenerator kpg=KeyPairGenerator.getInstance
("JCE");
            KeyPairGenerator kpg=KeyPairGenerator.getInstance("RSA");
            //KeyPairGenerator kpg=KeyPairGenerator.getInstance
("DiffieHellman");
            //KeyPairGenerator kpg=KeyPairGenerator.getInstance
("SHA1withRSA");
            //KeyPairGenerator kpg=KeyPairGenerator.getInstance
("DSA");

            kpg.initialize(1024);
            kp1 = kpg.generateKeyPair();
            priv1=kp1.getPrivate();
            publ=kp1.getPublic();

            // Saving Key Pair

            FileOutputStream fkeypairs = new FileOutputStream
(KeyFile1);
            ObjectOutputStream ookeypairs = new ObjectOutputStream
(fkeypairs);
            ookeypairs.writeObject(kp1);

            // Saving Public Key

            FileOutputStream fkeypubs = new FileOutputStream
(KeyFile2);
            ObjectOutputStream ookeypubs = new ObjectOutputStream
(fkeypubs);
            ookeypubs.writeObject(publ);

```

```

        // Saving the private Key
        FileOutputStream fkeyprivs = new FileOutputStream
(KeyFile3);
        ObjectOutputStream ookeyprivs = new ObjectOutputStream
(fkeyprivs);
        ookeyprivs.writeObject(priv1);

        ookeypairs.flush();
        ookeypubs.flush();
        ookeyprivs.flush();

        ookeypairs.close();
        ookeypubs.close();
        ookeyprivs.close();

    }

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

//Reading Keypair
public void KeyRead()
{
    try{
        // KeyPair Read
        System.out.println("Reading....");
        FileInputStream fis1=new FileInputStream(KeyFile1);
        ObjectInputStream ois1=new ObjectInputStream(fis1);

        kp2=(KeyPair)ois1.readObject();
        System.out.println("Your Key Pair:"+kp2);

        // Public Key Read

        FileInputStream fis2=new FileInputStream(KeyFile2);
        ObjectInputStream ois2=new ObjectInputStream(fis2);

        pub2=(PublicKey)ois2.readObject();
        System.out.println("Your Public Key :"+pub2);

        // Private Key Read

        FileInputStream fis3=new FileInputStream(KeyFile3);
        ObjectInputStream ois3=new ObjectInputStream(fis3);

        priv2=(PrivateKey)ois3.readObject();
        System.out.println("Your Private Key :"+priv2);

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

```

```

        public PrivateKey getprivatekey() throws FileNotFoundException,
IOException, ClassNotFoundException
        {
            return priv2;
        }

        public PublicKey getpublickey() throws FileNotFoundException,
IOException, ClassNotFoundException
        {
            return pub2;
        }

        public static void main(String args[]) throws
FileNotFoundException, IOException, ClassNotFoundException
        {
            keygen app = new keygen();
            app.setKeyName("BankKey");
            app.KeyGeneration();

            app.KeyRead();
            System.out.println("\n\nprivate here "+app.getprivatekey
());
            System.out.println("public here" +app.getpublickey());
        }
}

```

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.