AIM: To implement BlowFish Algorithm

BlowFish was developed by Bruce SchneierBlowfish is a symmetric block cipher.It encrypts 64 bit block with variable-length key, from 32 bits to 448 bits. It has two steps

Key Generation: This process converts the key up to 448 bits long to subkeys totaling 4168 bits.

Data Encryption: This process involves the iteration of a simple function 16 times.

Program

```
import javax.swing.*;
import java.security.*;
import javax.crypto.*;
import javax.crypto.spec.*;
import java.util.Random ;
class BlowFishAlgo
{
	byte[] skey = new byte[1000];
	String skeyString;
	static byte[] raw;
	String inputMessage,cipher1,plaintext;
	public BlowFishAlgo()
	{
		try
		{

			Random r = new Random();
			int num = r.nextInt(10000);
			String knum = String.valueOf(num);
			byte[] knumb = knum.getBytes();
			skey=getRawKey(knumb);
			skeyString = new String(skey);
			inputMessage=JOptionPane.showInputDialog(null,"Enter message to
encrypt");
			byte[] input = inputMessage.getBytes();
			byte[] cipher=encrypt(raw, input);
			String cipher1 = new String(cipher);
			//JOptionPane.showMessageDialog(null,"Encrypted Data "+"\n"+cipher1);
			JOptionPane.showMessageDialog (null, "Cipher Text is   " + cipher1,
"Encryption Process", JOptionPane.PLAIN_MESSAGE);
			byte[] dbyte= decrypt(raw,cipher);
			String plaintext = new String(dbyte);
			//JOptionPane.showMessageDialog(null,"Decrypted Data
"+"\n"+plaintext);
			JOptionPane.showMessageDialog (null, "Plain Text is  " + plaintext,
"Decryption Process", JOptionPane.PLAIN_MESSAGE);
```

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

      }
      private static byte[] getRawKey(byte[] seed) throws Exception
      {
            KeyGenerator kgen = KeyGenerator.getInstance("Blowfish");
            SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
            sr.setSeed(seed);
            kgen.init(128, sr); // 128, 256 and 448 bits may not be available
            SecretKey skey = kgen.generateKey();
            raw = skey.getEncoded();
            return raw;
      }
      private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception
      {
            SecretKeySpec skeySpec = new SecretKeySpec(raw, "Blowfish");
            Cipher cipherObj = Cipher.getInstance("Blowfish");
            cipherObj.init(Cipher.ENCRYPT_MODE, skeySpec);
            byte[] encrypted = cipherObj.doFinal(clear);
            return encrypted;
      }

      private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception
      {
            SecretKeySpec skeySpec = new SecretKeySpec(raw, "Blowfish");
            Cipher cipherObj1 = Cipher.getInstance("Blowfish");
            cipherObj1.init(Cipher.DECRYPT_MODE, skeySpec);
            byte[] decrypted = cipherObj1.doFinal(encrypted);
            return decrypted;
      }
      public static void main(String args[])
      {
            BlowFishAlgo blowObj = new BlowFishAlgo();
      }
}
```