```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Security.Cryptography;
using System.IO;
using System.Runtime.InteropServices; //Amit


namespace EncryptDecrypt
{
    //Amit Begin
    [Guid("9F2F180D-94A9-47e6-91CC-6BCFABD1DDEB")]
    [ClassInterface(ClassInterfaceType.None)]
    [ProgId("EncryptDecrypt.EncryptandDecrypts")]
    //Amit End
    [ComVisible(true)]

    public class EncryptandDecrypts: iEncryptDecrypt
    {
        public string encrypt(string encryptString)
        {
            string password = "0E1L2E3V4A5T6E";
            byte[] bytes = Encoding.Unicode.GetBytes(encryptString);
            using (Aes aes = Aes.Create())
            {
                Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(password,
new byte[13]
                {
          (byte) 73,
          (byte) 118,
          (byte) 97,
          (byte) 110,
          (byte) 32,
          (byte) 77,
          (byte) 101,
          (byte) 100,
          (byte) 118,
          (byte) 101,
          (byte) 100,
          (byte) 101,
          (byte) 118
                });
                aes.Key = rfc2898DeriveBytes.GetBytes(32);
                aes.IV = rfc2898DeriveBytes.GetBytes(16);
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    using (CryptoStream cryptoStream = new
CryptoStream((Stream)memoryStream, aes.CreateEncryptor(), CryptoStreamMode.Write))
                    {
                        cryptoStream.Write(bytes, 0, bytes.Length);
                        cryptoStream.Close();
                    }
                    encryptString = Convert.ToBase64String(memoryStream.ToArray());
                }
            }
            return encryptString;
```

```
        }

        public string Decrypt(string cipherText)
        {
            string password = "0E1L2E3V4A5T6E";
            cipherText = cipherText.Replace(" ", "+");
            byte[] buffer = Convert.FromBase64String(cipherText);
            using (Aes aes = Aes.Create())
            {
                Rfc2898DeriveBytes rfc2898DeriveBytes = new Rfc2898DeriveBytes(password,
new byte[13]
                {
          (byte) 73,
          (byte) 118,
          (byte) 97,
          (byte) 110,
          (byte) 32,
          (byte) 77,
          (byte) 101,
          (byte) 100,
          (byte) 118,
          (byte) 101,
          (byte) 100,
          (byte) 101,
          (byte) 118
                });
                aes.Key = rfc2898DeriveBytes.GetBytes(32);
                aes.IV = rfc2898DeriveBytes.GetBytes(16);
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    using (CryptoStream cryptoStream = new
CryptoStream((Stream)memoryStream, aes.CreateDecryptor(), CryptoStreamMode.Write))
                    {
                        cryptoStream.Write(buffer, 0, buffer.Length);
                        cryptoStream.Close();
                    }
                    cipherText = Encoding.Unicode.GetString(memoryStream.ToArray());
                }
            }
            return cipherText;
        }
    }
}
```