```
using System;
using System.Text;
using System.Security.Cryptography;

namespace Application.Utils
{
    public static class CryptographyUtil // AES-256
    {
        private static byte[] _key = {
            0x4A, 0x97, 0xE1, 0xE1, 0x24, 0x81, 0xD0, 0x61, 0xA9, 0xE8,
            0x4F, 0x82, 0x55, 0x11, 0x58, 0x0E, 0x0E, 0x64, 0x6E, 0x40,
            0x31, 0x1C, 0x31, 0xA9, 0xC7, 0x50, 0x12, 0x9C, 0x78, 0x04,
            0x10, 0xAF };
        private static byte[] _IV = {
            0x2F, 0xF7, 0xED, 0x6A, 0x5B, 0x27, 0x92, 0x23, 0x19, 0x49,
            0x64, 0x11, 0xD3, 0x91, 0x4E, 0xA5 };
        private static CipherMode _mode = CipherMode.CBC;
        private static PaddingMode _padding = PaddingMode.PKCS7;

        public static string Encode(string s, byte[] key, byte[] IV)
        {
            RijndaelManaged alg = new RijndaelManaged()
            {
                KeySize = 8 * key.Length,  // 256
                BlockSize = 8 * IV.Length, // 128
                Mode = _mode,
                Padding = _padding,
                Key = key,
                IV = IV
            };
            ICryptoTransform enc = alg.CreateEncryptor();
            byte[] d = Encoding.Unicode.GetBytes(s);
            byte[] c = enc.TransformFinalBlock(d, 0, d.Length);
            return Convert.ToBase64String(c);
        }

        public static string Encode(string s, byte[] key) =>
            Encode(s, key, _IV);

        public static string Encode(string s) =>
            Encode(s, _key, _IV);

        public static string Decode(string s, byte[] key, byte[] IV)
        {
            RijndaelManaged alg = new RijndaelManaged()
            {
                KeySize = 8 * key.Length,
                BlockSize = 8 * IV.Length,
                Mode = _mode,
                Padding = _padding,
                Key = key,
                IV = IV
            };
            ICryptoTransform dec = alg.CreateDecryptor();
            byte[] c = Convert.FromBase64String(s);
            byte[] d = dec.TransformFinalBlock(c, 0, c.Length);
            return Encoding.Unicode.GetString(d);
        }

        public static string Decode(string s, byte[] key) =>
            Decode(s, key, _IV);

        public static string Decode(string s) =>
            Decode(s, _key, _IV);
    }
}
```