

逆流成河

<	2019年12月						>
日	一	二	三	四	五	六	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	1	2	3	4	
5	6	7	8	9	10	11	

昵称: [口袋里的SKY](#)
园龄: [2年11个月](#)
粉丝: [15](#)
关注: [21](#)
[+加关注](#)

搜索

找找看

谷歌搜索

常用链接

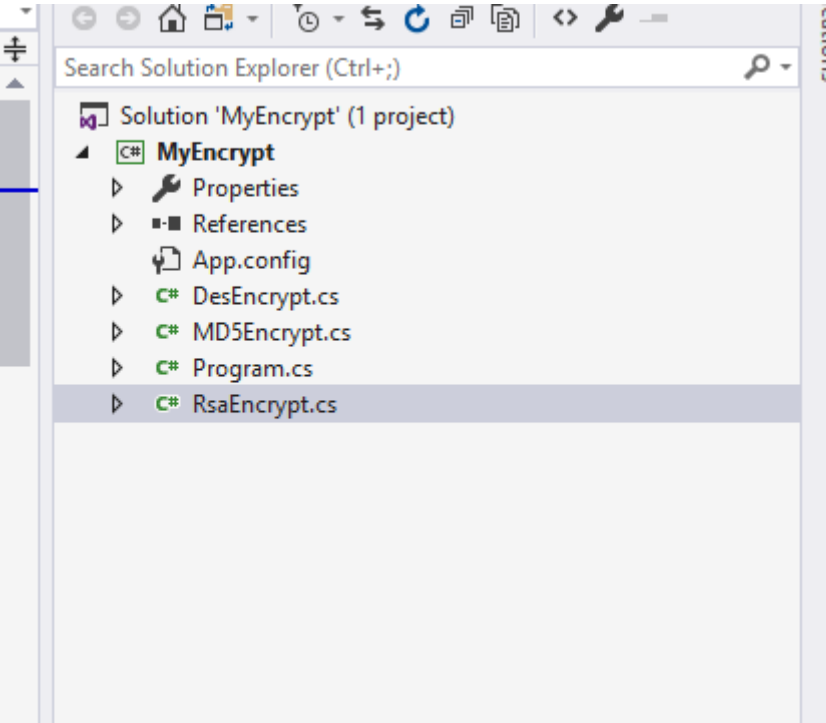
[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

随笔分类

[C#\(3\)](#)
[c#高级课程\(29\)](#)
[C#基础\(1\)](#)
[EF](#)
[EF实体框架\(1\)](#)
[JS/JQuery/CSS](#)
[MVC\(1\)](#)
[代码管理工具\(1\)](#)
[第三方平台的应用\(1\)](#)
[内存\(1\)](#)

[博客园](#) [首页](#) [新随笔](#) [新文章](#) [联系](#) [订阅](#) [RSS](#) [管理](#)
posts - 187,comments - 11,trackbacks - 0

Encrypt 加密



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace MyEncrypt
{
    /// <summary>
```

[其他\(4\)](#)
[设计模式\(1\)](#)
[实例代码和各种帮助类\(9\)](#)
[数据格式\(3\)](#)
[数据结构\(4\)](#)
[数据库\(4\)](#)
[数据类型转换\(3\)](#)
[通信协议\(3\)](#)

随笔档案

[2019年6月\(2\)](#)
[2019年5月\(2\)](#)
[2018年5月\(1\)](#)
[2018年4月\(16\)](#)
[2018年3月\(24\)](#)
[2018年2月\(4\)](#)
[2018年1月\(19\)](#)
[2017年12月\(2\)](#)
[2017年11月\(6\)](#)
[2017年10月\(5\)](#)
[2017年9月\(4\)](#)
[2017年8月\(2\)](#)
[2017年7月\(6\)](#)
[2017年6月\(15\)](#)
[2017年5月\(27\)](#)
[2017年4月\(16\)](#)
[2017年3月\(25\)](#)
[2017年2月\(11\)](#)

最新评论

1. Re:2017-6-6 ajax 完整+三级联动赞

--我之姓冠你之名

2. Re:2017-6-8 上传图片加水印, 验证码制作

别问我你是谁 !!! 我是说你写博客 图文并茂啊啊啊

--我之姓冠你之名

3. Re:2017-6-8 上传图片加水印, 验证码制作

@ 我之姓冠你之名你也学.net??...

--口袋里的SKY

4. Re:2017-4-17 类库 委托 通用变量 is和as

@ 霸道小豆丁从ado.net 开始的有的例子...

--口袋里的SKY

5. Re:2017-4-17 类库 委托 通用变量 is和as

这个还用例子吗? is 是判断是什么类型 返回布尔值 as就是把某种类型转换成目标类型

```

/// DES AES Blowfish
/// 对称加密算法的优点是速度快,
/// 缺点是密钥管理不方便, 要求共享密钥。
/// 可逆对称加密 密钥长度8
/// </summary>
public class DesEncrypt
{
    ///8位长度
    private static string KEY = "ruanmoul";
    private static byte[] rgbKey =
        ASCIIEncoding.ASCII.GetBytes(KEY.Substring(0, 8));
    private static byte[] rgbIV =
        ASCIIEncoding.ASCII.GetBytes(KEY.Insert(0,
        "w").Substring(0, 8));

    /// <summary>
    /// DES 加密
    /// </summary>
    /// <param name="strValue"></param>
    /// <returns></returns>
    public static string Encrypt(string strValue)
    {
        DESCryptoServiceProvider dsp = new
        DESCryptoServiceProvider();
        using (MemoryStream memStream = new
        MemoryStream())
        {
            CryptoStream crypStream = new
            CryptoStream(memStream, dsp.CreateEncryptor(rgbKey,
            rgbIV), CryptoStreamMode.Write);
            StreamWriter sWriter = new
            StreamWriter(crypStream);
            sWriter.Write(strValue);
            sWriter.Flush();
            crypStream.FlushFinalBlock();
            memStream.Flush();
            return
            Convert.ToBase64String(memStream.GetBuffer(), 0,
            (int)memStream.Length);
        }
    }

    /// <summary>
    /// DES解密
    /// </summary>
    /// <param name="EncValue"></param>
    /// <returns></returns>
    public static string Decrypt(string EncValue)
    {
        DESCryptoServiceProvider dsp = new
        DESCryptoServiceProvider();
        byte[] buffer =
        Convert.FromBase64String(EncValue);

        using (MemoryStream memStream = new

```

--口袋里的SKY

阅读排行榜

1. [Encrypt 加密\(2676\)](#)
2. [TortoiseHg 学习笔记 \(转\) \(1976\)](#)
3. [调用第三方接口 \(高德地图\) \(1533\)](#)
4. [vs2017 的激活密钥\(1317\)](#)
5. [C#中Dictionary的用法\(转\)\(393\)](#)

评论排行榜

1. [2017-4-17 类库 委托 通用变量 is 和as\(3\)](#)
2. [2017-6-8 上传图片加水印, 验证码制作\(3\)](#)
3. [2017-5-4 进程\(2\)](#)
4. [2017-6-6 ajax 完整+三级联动\(1\)](#)
5. [2017-2-20 C#基础 运算符\(1\)](#)

推荐排行榜

1. [2017-3-10 SQL server 数据库 T-SQL语句\(1\)](#)
2. [2017-4-21 ADO.NET的六大对象\(1\)](#)
3. [2017-4-20 ADO.NET 实体类 数据访问类\(1\)](#)
4. [2017-4-23 心情\(1\)](#)
5. [2017-4-24 WinForm开发基础、窗体的属性\(1\)](#)

```
MemoryStream())
{
    CryptoStream crypStream = new
CryptoStream(memStream, dsp.CreateDecryptor(rgbKey,
rgbIV), CryptoStreamMode.Write);
    crypStream.Write(buffer, 0,
buffer.Length);
    crypStream.FlushFinalBlock();
    return
ASCIIEncoding.UTF8.GetString(memStream.ToArray());
}
}
}
```



```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace MyEncrypt
{
    /// <summary>
    /// 不可逆加密
    /// 1 防止被篡改
    /// 2 防止明文存储
    /// 3 防止抵赖, 数字签名
    /// </summary>
    public class MD5Encrypt
    {
        #region MD5
        /// <summary>
        /// MD5加密,和动网上的16/32位MD5加密结果相同,
        /// 使用的UTF8编码
        /// </summary>
        /// <param name="strSource">待加密字符串</param>
        /// <param name="length">16或32值之一,其它则采用.net
        默认MD5加密算法</param>
        /// <returns>加密后的字符串</returns>
        public static string Encrypt(string source, int
length = 32)//默认参数
        {
            HashAlgorithm provider =
CryptoConfig.CreateFromName("MD5") as HashAlgorithm;
            if (string.IsNullOrEmpty(source)) return
string.Empty;

            byte[] bytes =
```

```

Encoding.UTF8.GetBytes(source); //
Encoding.ASCII.GetBytes(source);
    byte[] hashValue =
provider.ComputeHash(bytes);
    StringBuilder sb = new StringBuilder();
    switch (length)
    {
        case 16: // 16位密文是32位密文的9到24位字符
            for (int i = 4; i < 12; i++)

sb.Append(hashValue[i].ToString("x2"));
                break;
        case 32:
            for (int i = 0; i < 16; i++)
            {

sb.Append(hashValue[i].ToString("x2"));
                }
                break;
        default:
            for (int i = 0; i < hashValue.Length;
i++)
            {

sb.Append(hashValue[i].ToString("x2"));
                }
                break;
            }
        return sb.ToString();
    }
}
#endregion MD5
}

```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MyEncrypt
{
    /// <summary>
    /// 1: MD5 不可逆加密
    /// 2: Des 对称可逆加密
    /// 3: RSA 非对称可逆加密
    /// 4: 数字证书 SSL
    /// </summary>
    class Program
    {
        static void Main(string[] args)

```

```
{  
    try  
    {  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456小杨中交一公局第  
六工程有限公司"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456小夏中交一公局第  
六工程有限公司"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456小吴中交一公局第  
六工程有限公司"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456李小中交一公局第  
六工程有限公司"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456小李中交一公局第  
六工程有限公司"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("123456小李"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("1"));  
        //不可逆  
        Console.WriteLine(MD5Encrypt.Encrypt("1冉  
娃娃是.net高级班的一名vip学员,他是四川的小伙子,目前好像还没结婚  
47926363838753576475637566573654756735438547385434"));  
        Console.WriteLine(MD5Encrypt.Encrypt("2冉  
娃娃是.net高级班的一名vip学员,他是四川的小伙子,目前好像还没结婚  
47926363838753576475637566573654756735438547385434"));  
        Console.WriteLine(MD5Encrypt.Encrypt("3冉  
娃娃是.net高级班的一名vip学员,他是四川的小伙子,目前好像还没结婚  
47926363838753576475637566573654756735438547385434"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("1"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("xiaoduirensheng1234  
55"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("kuaileyangguang1234  
55"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("haha123455"));  
  
        Console.WriteLine(MD5Encrypt.Encrypt("haha123456"));  
        Console.WriteLine(MD5Encrypt.Encrypt("张三  
李四"));  
        Console.WriteLine(MD5Encrypt.Encrypt("张三  
李四"));  
  
        //可逆对称加密  
        string desEn = DesEncrypt.Encrypt("王殃  
殃");  
        string desDe = DesEncrypt.Decrypt(desEn);  
        string desEn1 = DesEncrypt.Encrypt("张三李  
四");
```

```

        string desDel =
DesEncrypt.Decrypt(desEn1);

        //非对称加密
        //string publicKey = "";
        //string privateKey = "";
        //string rsaEn =
RsaEncrypt.Encrypt("netnetnetnetnetnetnetne", out
publicKey, out privateKey);
        //string rsaDe =
RsaEncrypt.Decrypt(rsaEn, privateKey);

        KeyValuePair<string, string>
publicPrivate = RsaEncrypt.GetKeyPair();
        string rsaEn1 = RsaEncrypt.Encrypt("net",
publicPrivate.Key);
        string rsaDel =
RsaEncrypt.Decrypt(rsaEn1, publicPrivate.Value);
        //加密key 解密key
        //公开加密key, 接受加密消息, 因为只有我一个人能解
密
        //公开解密key, 用于签名, 表明数据一定是我发的,
因为只有我有加密的key
        //公钥私钥只是跟公开与否有关
    }
    catch (Exception ex)
    {
        Console.WriteLine(ex.Message);
    }
    Console.Read();
}
}

```



```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;

namespace MyEncrypt
{
    /// <summary>
    /// RSA ECC
    /// 可逆非对称加密
    /// 非对称加密算法的优点是密钥管理很方便, 缺点是速度慢。
    /// </summary>
    public class RsaEncrypt

```

```

{

    /// <summary>
    /// publicKey,privateKey
    /// </summary>
    /// <returns></returns>
    public static KeyValuePair<string, string>
GetKeyPair()
    {
        RSACryptoServiceProvider RSA = new
RSACryptoServiceProvider();
        string publicKey = RSA.ToXmlString(false);
        string privateKey = RSA.ToXmlString(true);
        return new KeyValuePair<string, string>
(publicKey, privateKey);
    }

    /// <summary>
    /// 加密: 内容+公钥
    /// </summary>
    /// <param name="content"></param>
    /// <param name="publicKey"></param>
    /// <returns></returns>
    public static string Encrypt(string content,
string publicKey)
    {
        RSACryptoServiceProvider rsa = new
RSACryptoServiceProvider();
        rsa.FromXmlString(publicKey);
        UnicodeEncoding ByteConverter = new
UnicodeEncoding();
        byte[] DataToEncrypt =
ByteConverter.GetBytes(content);
        byte[] resultBytes =
rsa.Encrypt(DataToEncrypt, false);
        return Convert.ToBase64String(resultBytes);
    }

    /// <summary>
    /// 加密
    /// </summary>
    /// <param name="content"></param>
    /// <param name="publicKey">返回公钥</param>
    /// <param name="privateKey">返回密钥</param>
    /// <returns>加密后结果</returns>
    public static string Encrypt(string content, out
string publicKey, out string privateKey)
    {
        RSACryptoServiceProvider rsaProvider = new
RSACryptoServiceProvider();
        publicKey = rsaProvider.ToXmlString(false);
        privateKey = rsaProvider.ToXmlString(true);

        UnicodeEncoding ByteConverter = new
UnicodeEncoding();

```

```
byte[] DataToEncrypt =
ByteConverter.GetBytes(content);
byte[] resultBytes =
rsaProvider.Encrypt(DataToEncrypt, false);
return Convert.ToBase64String(resultBytes);
}

/// <summary>
/// 解密 内容+私钥
/// </summary>
/// <param name="content"></param>
/// <param name="privateKey"></param>
/// <returns></returns>
public static string Decrypt(string content,
string privateKey)
{
byte[] dataToDecrypt =
Convert.FromBase64String(content);
RSACryptoServiceProvider RSA = new
RSACryptoServiceProvider();
RSA.FromXmlString(privateKey);
byte[] resultBytes =
RSA.Decrypt(dataToDecrypt, false);
UnicodeEncoding ByteConverter = new
UnicodeEncoding();
return ByteConverter.GetString(resultBytes);
}
}
```



分类: [c#高级课程](#)

好文要顶

关注我

收藏该文



[口袋里的SKY](#)

[关注 - 21](#)

[粉丝 - 15](#)

[+加关注](#)

0

0

« 上一篇: [Lottery彩票中奖](#)

» 下一篇: [Encrypt 加密 \(图解\)](#)

posted on 2018-03-20 08:50 [口袋里的SKY](#) 阅读(2676) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#) [网站首页](#)。

[【推荐】超50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库](#)

[【推荐】腾讯云热门云产品限时秒杀，爆款1核2G云服务器99元/年！](#)

[【推荐】阿里云双11返场来袭，热门产品低至一折等你来抢！](#)

[【活动】京东云服务器 云主机低于1折，低价高性能产品备战双11](#)

[【活动】ECUG For Future 技术者的年度盛会（杭州，1月4-5日）](#)

相关博文：

- [encrypt.hencrypt.cpp](#)
- [Encrypt（加密解密）](#)
- [js 加密方法Encrypt](#)
- [php密码对称encrypt加密](#)
- [著名加密库收集 Encrypt](#)
- » [更多推荐...](#)

[精品问答：精品问答：Python 技术 1000 问](#)

最新 IT 新闻：

- [被骂垃圾货，却卖出8000万副，干翻国外大牌！这个产品杀手凭什么？](#)
- [马云谈2019太难了：光昨天我就收到5个朋友借钱电话](#)
- [外媒称苹果开放漏洞悬赏计划 至少奖励100万美金](#)
- [“玉兔二号”成为人类月面工作时间最长月球车](#)
- [1：2，李世石最后一战被AI击败，唯一战胜过AlphaGo的人退役了](#)
- » [更多新闻...](#)

历史上的今天：

2017-03-20 [2017-3-20 HTML 基础知识](#)

Copyright © 2019 口袋里的SKY

Powered by .NET Core 3.1.0 on Linux Powered By[博](#)

[客园](#) 模板提供：[沪江博客](#)