用C#实现的数据加密(一)对称加密算法 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/219/2021_2022__E7_94_A8C_23_E5_AE_9E_E7_8E_c67_219879.htm 计算机等级考试训练软件《百宝箱》用C#实现的数据加密(一) 对称加密算法以下是关于对称加密算法的C#实现代码，大家可以根据需要更改不同的算法，文中以Rijndael算法为例：using System.using System.IO.using System.Security.Cryptography.using System.Text.namespace DataCrypto{/// /// 对称加密算法类/// public class SymmetricMethod{private SymmetricAlgorithm mobjCryptoService.private string Key./// /// 对称加密类的构造函数/// public SymmetricMethod(){mobjCryptoService = new RijndaelManaged().Key = "Guz(%amp.fvHUFCy76*h %(HilJ$lhj!y6amp.95GUY86GfghUb#er57HBh(u %g6HJ($jhWk7&amp.!hg4ui %$hjk".mobjCryptoService.GenerateIV().byte[] bytTemp = mobjCryptoService.IV.int IVLength = bytTemp.Length.if (sTemp.Length > IVLength)sTemp = sTemp.Substring(0, IVLength).else if (sTemp.Length sTemp = sTemp.PadRight(IVLength,).return ASCIIEncoding.ASCII.GetBytes(sTemp).}/// /// 加密方法/// /// 待加密的串/// 经过加密的串public string Encrypto(string Source){byte[] bytIn = UTF8Encoding.UTF8.GetBytes(Source).MemoryStream ms = new MemoryStream().mobjCryptoService.Key = GetLegalKey().mobjCryptoService.IV =

GetLegalIV().ICryptoTransform encrypto = mobjCryptoService.CreateEncryptor().CryptoStream cs = new CryptoStream(ms, encrypto, CryptoStreamMode.Write).cs.Write(bytIn, 0, bytIn.Length).cs.FlushFinalBlock().ms.Close().byte[] bytOut = ms.ToArray().return Convert.ToBase64String(bytOut).}/// /// 解密方法/// /// 待解密的串/// 经过解密的串public string Decrypto(string Source){byte[] bytIn = Convert.FromBase64String(Source).MemoryStream ms = new MemoryStream(bytIn, 0, bytIn.Length).mobjCryptoService.Key = GetLegalKey().mobjCryptoService.IV = GetLegalIV().ICryptoTransform encrypto = mobjCryptoService.CreateDecryptor().CryptoStream cs = new CryptoStream(ms, encrypto, CryptoStreamMode.Read).StreamReader sr = new StreamReader(cs).return sr.ReadToEnd().}}}