链滴

Java 与 C# 互通的 AES 加密，加密解密结果一致

作者：chinaYoung
原文链接：https://ld246.com/article/1535687542428
来源网站：链滴

在实际项目中，遇到Java系统与.Net系统进行对接；由Java出发的加密数据需要在.Net中解密进行下步的操作，在网上查阅了很多资料，总结出以下两个版本通用的方法，把加密字节通过Base64再加一次，解决传输乱码的问题保证加密解密结果完全一致。

JAVA版本

AESUtils工具类

```
package www.**.**.util;

import sun.misc.BASE64Decoder;
import sun.misc.BASE64Encoder;

import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

/**
 * @author yangps.
 * @version 2018-8-28下午 14:23
 */public class AESUtils {
    //实际的加密解密操作
    private static String Operation(String src, String key, int mode) throws Exception {
        if (key==null) {
            return "Key不能为空";
        }
        if (key.length()!=16) {
            return "Key需要16位长度";
        }
        String result = "";
        byte[] raw = key.getBytes("utf-8");
        SecretKeySpec keySpec = new SecretKeySpec(raw, "AES");
        Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
        if (mode == Cipher.ENCRYPT_MODE) {
            cipher.init(mode, keySpec);
            byte[] encrypted = cipher.doFinal(src.getBytes("utf-8"));
            //将+替换为%2B
      result = new BASE64Encoder().encode(encrypted).replace("+", "%2B");
        } else {
            cipher.init(mode, keySpec);
            //将%2B替换为+
      src = src.replace("%2B", "+");
            byte[] encrypted = cipher.doFinal(new BASE64Decoder().decodeBuffer(src));
            result = new String(encrypted, "utf-8");
        }

        return result;
    }

    /**
 * 加密 * * @param src
  * @return
  * @throws Exception
```

```
 */  public static String Encrypt(String src, String key) throws Exception {
        return Operation(src, key, Cipher.ENCRYPT_MODE);
    }

    /**
 * 解密 * * @param src
  * @return
  * @throws Exception
 */  public static String Decrypt(String src, String key) throws Exception {
        return Operation(src, key, Cipher.DECRYPT_MODE);
    }

}
```

C#版本

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Security.Cryptography;
namespace Common
{
    public class Encry
    {
        public Encry() { }

        #region AES加密

        public static string Encrypt(string toEncrypt, string key)
        {

            if (key == null || key.Length != 16) {
                return "key不能为空并且需要16位长度";
            }
            byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
            byte[] toEncryptArray = UTF8Encoding.UTF8.GetBytes(toEncrypt);

            RijndaelManaged rDel = new RijndaelManaged();
            rDel.Key = keyArray;
            rDel.Mode = CipherMode.ECB;
            rDel.Padding = PaddingMode.PKCS7;

            ICryptoTransform cTransform = rDel.CreateEncryptor();
            byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray
Length);
            //将+替换为%2B
            string resultArrayAfter = Convert.ToBase64String(resultArray, 0, resultArray.Length);
            resultArrayAfter = resultArrayAfter.Replace("+", "%2B");

            return resultArrayAfter;
        }
        #endregion AES加密
```

```
        #region AES解密

        public static string Decrypt(string toDecrypt, string key)
        {
            //将%2B替换为+
            toDecrypt = toDecrypt.Replace("%2B", "+");
            byte[] keyArray = UTF8Encoding.UTF8.GetBytes(key);
            byte[] toEncryptArray = Convert.FromBase64String(toDecrypt);

            RijndaelManaged rDel = new RijndaelManaged();
            rDel.Key = keyArray;
            rDel.Mode = CipherMode.ECB;
            rDel.Padding = PaddingMode.PKCS7;

            ICryptoTransform cTransform = rDel.CreateDecryptor();
            byte[] resultArray = cTransform.TransformFinalBlock(toEncryptArray, 0, toEncryptArray
Length);

            return UTF8Encoding.UTF8.GetString(resultArray);
        }

        #endregion AES解密
    }
}
```