

数据编解码与加解密

2019年3月8日 9:57

■ 编码与解码

编码是将信息从一种格式（人能识别）转换为另一种形式（计算机能识别）的过程，反之成为解码。

字符集与字符编码 <<http://www.cnblogs.com/skynet/archive/2011/05/03/2035105.html>>

理解字符集和编码，常见字符集名称：ASCII、GB2312、BIG5、GB18030 和 Unicode 字符集。

```
int unicode = char.Parse("零");  
var unicodeCodeString = Convert.ToString(unicode, 2);
```

微软提供的主要类：System.Text.Encoding

```
var bytes = BitConverter.GetBytes(unicode);  
UTF32Encoding encoding = new UTF32Encoding();  
var str = encoding.GetString(bytes);
```

关于 Base64 编码

Base64编码原理 <<https://www.jianshu.com/p/8aa883943ab3>>

```
byte[] bytes = Encoding.Default.GetBytes(str);  
string baseString = Convert.ToBase64String(bytes);  
  
byte[] bytes = Convert.FromBase64String(baseString);  
var str = Encoding.Default.GetString(bytes);
```

■ 哈希散列摘要算法

Hash，一般翻译做散列，音译为哈希，是把任意长度的输入通过散列算法变换成固定长度的输出，该输出就是散列值。简单的说就是一种将任意长度的消息压缩到某一固定长度的消息摘要的算法或者函数，有时也叫摘要算法。

目前主流的 Hash 算法有：MD4、MD5 和 SHA 系列。

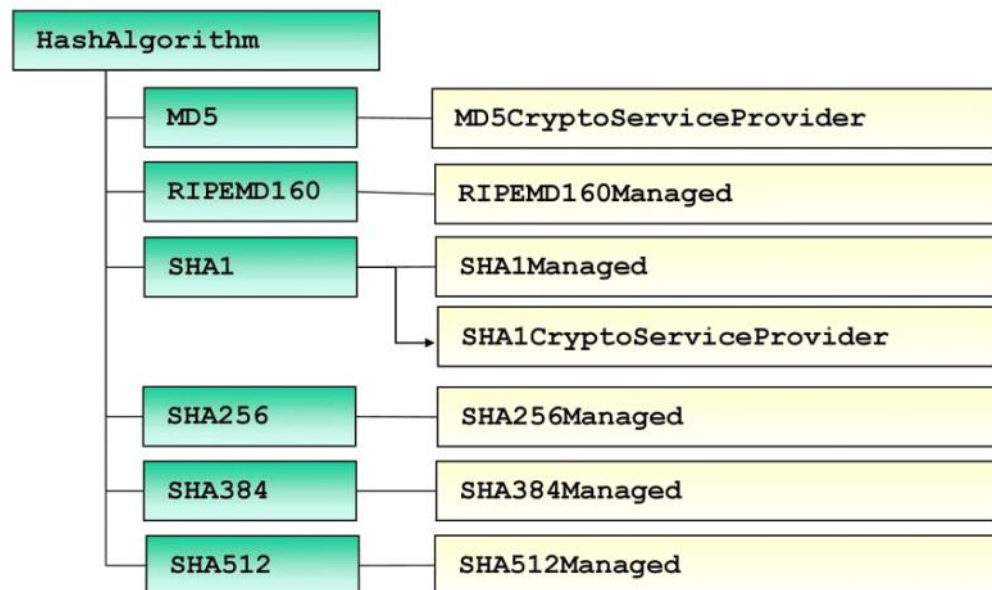
什么是Hash算法 <<https://www.jianshu.com/p/bba9b61b80e7>>

SHA家族有5个算法：SHA-1，SHA-224，SHA-256，SHA-384 和 SHA-512，其中后 4 者总称 SHA2 算法。

用途：文件校验、数字签名 和 鉴权协议。

浅谈hash一致性算法 <<https://www.jianshu.com/p/6f13156573f4>>

微软实现来自 <<https://github.com/dotnet/corefx/tree/master/src/System.Security.Cryptography.Algorithms>>



```
public byte[] ComputeHash(byte[] buffer)

string strContent = "www.xcode.me;video.xcode.me";

byte[] bytes = Encoding.Default.GetBytes(strContent);

//var sha1 = new System.Security.Cryptography.SHA1CryptoServiceProvider();

var sha1 = System.Security.Cryptography.SHA1.Create();

byte[] result = sha1.ComputeHash(bytes);

string sha1HashString = BitConverter.ToString(result);
```

HMAC 是具密钥的哈希算法，以一个密钥和一个消息为输入，生成摘要作为输出。

消息 + 密钥 + 算法 ==》 输出摘要

```
string strText = "This is .net core website";

string strKey = "www.xcode.me";

HMACSHA1 hmacsha1 = new HMACSHA1(Encoding.UTF8.GetBytes(strKey));
```

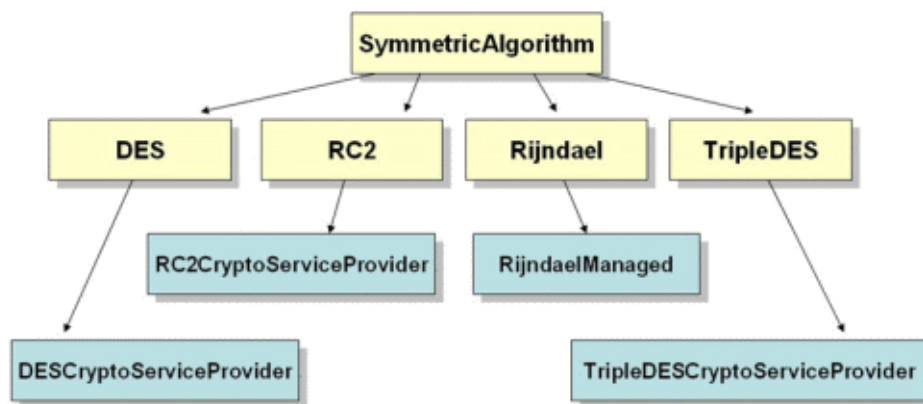
```
byte[] bytes = hmacsha1.ComputeHash(Encoding.UTF8.GetBytes(strText));  
  
string hmacSha1String= BitConverter.ToString(bytes);  
  
string hmacSha1BaseString = Convert.ToBase64String(bytes);
```

校验摘要：就是将原文进行同样的算法后比较摘要的过程。

■ 对称加密算法

采用单钥密码系统的加密方法，同一个密钥可以同时用作信息的加密和解密，这种加密方法称为对称加密，也称为单密钥加密。

常见的对称加解密算法：Aes、RC2、DES、TripleDES 和 Rijndael 算法。

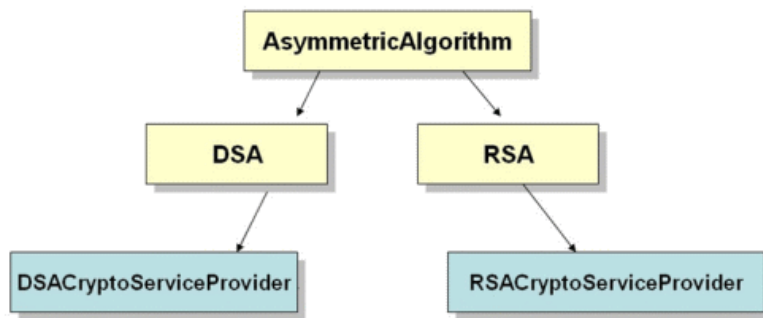


[TripleDESCryptoServiceProvider使用示例](#)

来自简书的TripleDES加密示例 <<https://www.jianshu.com/p/0781a6f21305>>

■ 非对称加密算法

非对称加密算法需要两个密钥：公开密钥（publickey）和私有密钥（privatekey）。公开密钥与私有密钥是一对，如果用公开密钥对数据进行加密，只有用对应的私有密钥才能解密；如果用私有密钥对数据进行加密，那么只有用对应的公开密钥才能解密。因为加密和解密使用的是两个不同的密钥，所以这种算法叫作非对称加密算法。非对称加密算法实现机密信息交换的基本过程是：甲方生成一对密钥并将其中的一把作为公用密钥向其它方公开；得到该公用密钥的乙方使用该密钥对机密信息进行加密后再发送给甲方；甲方再用自己保存的另一把专用密钥对加密后的信息进行解密。

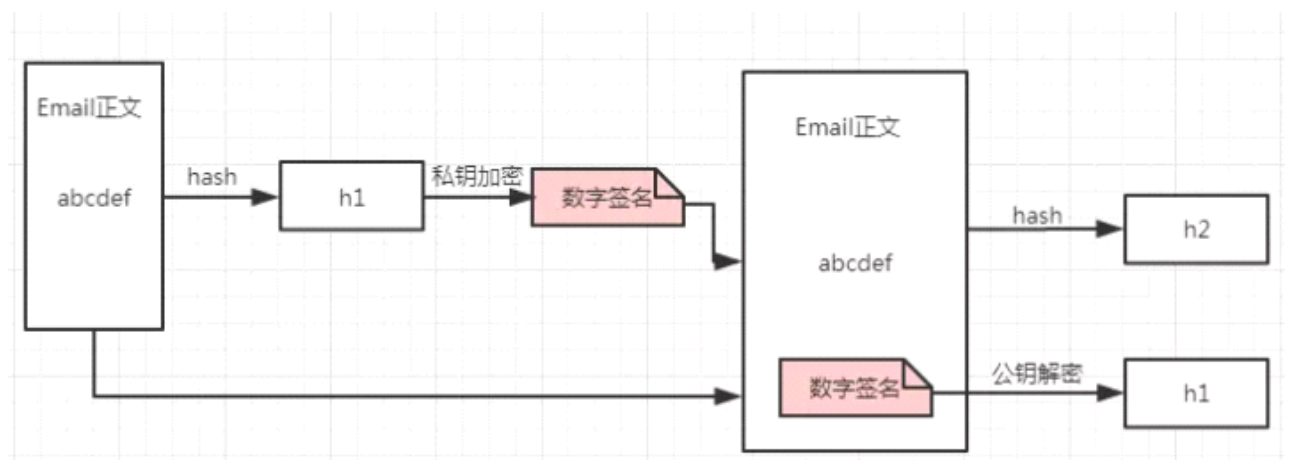


[RSACryptoServiceProvider](#)

开源的便捷加密解密库 <<https://github.com/myloveCc/NETCore.Encrypt>>

■ 数字签名

只有信息的发送者才能产生的别人无法伪造的一段数字串，这段数字串同时也是对信息的发送者发送信息真实性的一个有效证明。



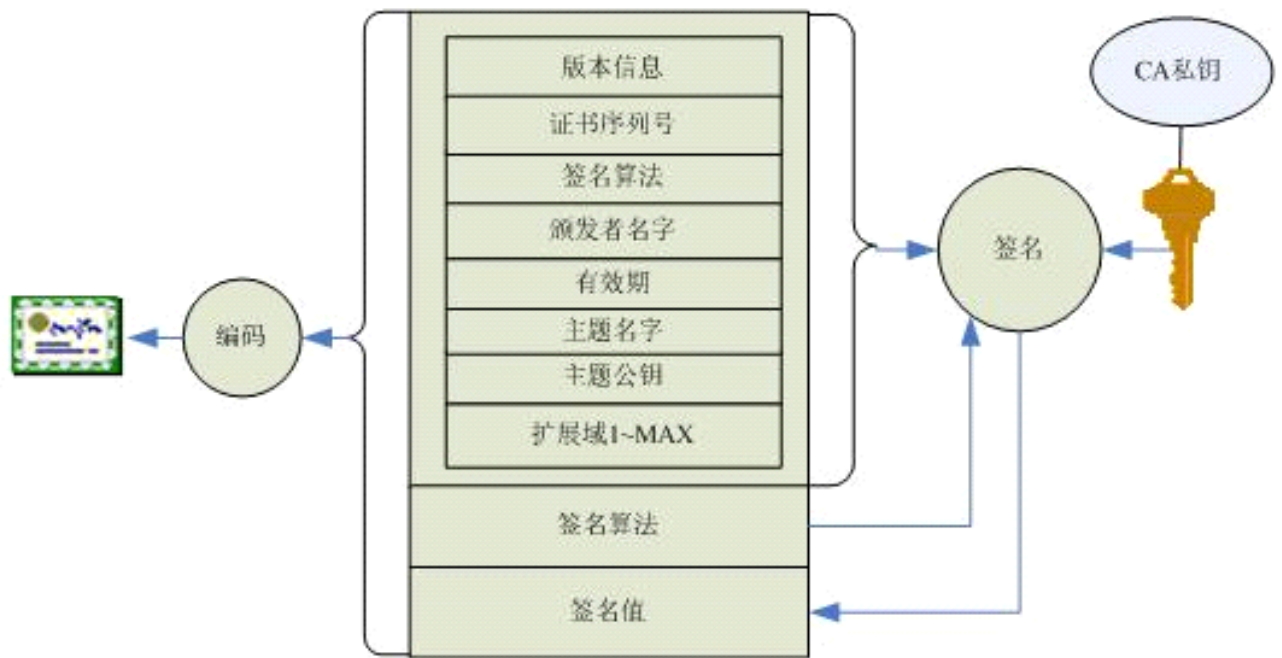
屏幕剪辑的捕获时间: 2019-03-14 22:18

数字签名是非对称密钥加密技术与数字摘要技术的应用。

来自 <<https://www.jianshu.com/p/090e35989501>>

■ 数字证书

数字证书就是互联网通讯中标志通讯各方身份信息的一串数字，提供了一种在网络上验证通信实体身份的方式，它是由权威机构颁发的，人们可以在网上用它来识别对方的身份。



MakeCert 创建证书工具

来自 <<https://docs.microsoft.com/en-us/windows/desktop/SecCrypto/makecert>>

微软提供的数字证书处理类

来自 <<https://docs.microsoft.com/en-us/dotnet/api/system.security.cryptography.x509certificates?view=netcore-3.0>>

■ 扩展阅读内容

来自 <<https://www.cnblogs.com/jams742003/category/241179.html>>

推荐《微软.NET程序的加密与解密》和《.NET安全揭秘》图书

