在ASP.Net中实现RSA加密的方法

转载 2013-11-07 作者： 我要评论

这篇文章介绍了在ASP.Net中实现RSA加密的方法，有需要的朋友可以参考一下

在我们实际运用中，加密是保证数据安全的重要手段。以前使用ASP时，对数据加密可以使用MD5和SHA1算法，这两种算法虽然快捷有效，但是无法对通过它们加密的密文进行反运算，即是解密。因此需要解密数据的场合，这两种方法就不太适合了。当然你也可以自己编写适用的加密和解密程序，不过这对编写者的数学水平有很高的要求，一般人是很难做到的。

现在，随着ASP.Net的推出，彻底改变了以前ASP下的编程模式。我们能够利用.Net Framework中的类提供的加密服务来保证数据安全。目前应用较为广泛的加密方法是使用RSA算法进行加密。在.Net Framework中与RSA加密算法相关的类主要有两个：RSA 类和RSACryptoServiceProvider 类。按照MSDN的说法RSA 类是"表示 RSA 算法的所有实现均从中继承的基类"，而RSACryptoServiceProvider 类是"使用加密服务提供程序 (CSP) 提供的 RSA 算法的实现执行不对称加密和解密"。另外，"表示 RSA 算法的标准参数"的RSAParameters 结构也是很重要的，它保存了RSA算法的参数。

由于介绍RSA算法原理的文章或书籍比较多，大家可以参阅一下，在此就不复述了。下面着重介绍一下如何在ASP.Net中实现RSA加密。

大家感兴趣的内容
1 JAVA正则表达式 Pattern和Matche
2 asp.net(c#)网页跳转七种方法小结
3 未将对象引用设置到对象的实例 (
4 未能加载文件或程序集“XXX”或它
5 asp.net “服务器应用程序不可用”
6 ASP.NET中的几种弹出框提示基本实
7 asp.net gridview 72般绝技
8 asp.net生成Excel并导出下载五种
9 asp.net UpdatePanel的简单用法
10 ASP.NET对路径"xxxxx"

RSA参数的产生：RSA参数的类型就是上面提到的RSAParameters 结构，查阅MSDN可知其包含了D、DP、DQ、Exponent、InverseQ、Modulus、P、Q八个字段。加密时仅需要Exponent和Modulus两个值，可看成公钥。解密时所有字段都需要，可看成私钥。下面这段程序显示了如何产生RSA两个参数：

代码如下: 复制代码

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
RSAParameters rsaParamsExcludePrivate=rsa.ExportParameters(false);
RSAParameters rsaParamsIncludePrivate=rsa.ExportParameters(true);
```

RSACryptoServiceProvider类的ExportParameters(bool)方法用于导出RSA参数，true表示导出上述八个字段的"私钥"，false表示导出"公钥"。

使用RSA参数进行加密解密：这一步需要把上面两个参数导入到RSACryptoServiceProvider类对象中，再用它对数据进行加密。如下面的代码所示，我们可以写一个函数来完成加密过程：

代码如下: 复制代码

```
Public byte [ ] RSAEncrypt ( byte [ ] b)
{
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
rsa.ImportParameters(rsaParamsExcludePrivate); //导入公钥
byte [] EncryptedData=rsa.Encrypt(DataToEncrypt,false);
return EncryptedData;
}
```

解密时只要把rsa.ImportParameters(rsaParamsExcludePrivate)换成rsa.ImportParameters(rsaParamsExcludePrivate)，再把Encrypt换成Decrypt就行了。

保存和加载RSA参数：RSA参数可以保存为XML格式，下面代码说明了如何保存和加载（只列出了关键部分）

保存:

代码如下: 复制代码

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
StreamWriter writer=new StreamWriter(@"d:\PublicAndPrivateKey.xml");
string PPKeyXml=rsa.ToXmlString(true);//保存私钥 writer.Write(PPKeyXml);
writer.Close();
writer=new StreamWriter(@"d:\PublicKey.xml");
string PKeyXml=rsa.ToXmlString(false);//保存公钥
writer.Write(PKeyXml);
writer.Close();
```

读取：

代码如下: 复制代码

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider();
StreamReader reader=new StreamReader(@"d:\PublicKey.xml");
string PKey=reader.ReadToEnd();
rsa.FromXmlString(PKey);
reader.Close();
StreamReader reader=new StreamReader(@"d:\PublicAndPrivateKey.xml");
string PPKey=reader.ReadToEnd();
reader.Close();
```

ToXmlString和ExportParameters方法类似，false表示保存"公钥"，true表示保存"私钥"。

您可能感兴趣的文章:

- ASP.net中md5加密码的方法
- asp.net TripleDES加密、解密算法
- asp.net下常用的加密算法MD5、SHA-1应用代码

最近更新的内容

常用在线小工具