基于 MOSS 的安全 Web 电子邮件客户端的设计与实现①

Design and Implementation of Secure Web E-mail Client Based on MOSS Protocol

张学旺 （重庆邮电大学软件学院 重庆 400065）

摘要:MOSS 安全电子邮件协议提供对 MIME 对象进行数字签名和数据加密的服务。本文基于 MOSS 协议设计实现了一个 Web 安全电子邮件客户端系统,描述了系统结构、邮件发送和邮件接收的全过程,重点分析了 MOSS 协议的数字签名和数据加密服务在系统中的设计与实现。

关键词:MOSS 数字签名 数据加密 安全电子邮件客户端

1 引言

为了扩充电子邮件的安全性能,IETF 起草了一些相关的规范,如 PEM、S/MIME、MOSS 等。MOSS 协议(MIME Object Security Services,即 MIME 对象安全服务)结合了 PEM 和 MIME 的特性,在应用层提供对 MIME 对象进行签名和加密的服务;并综合使用了非对称密码系统和对称密码系统,其中前者用于支持数字签名和密钥管理服务,后者用于支持数据加密服务。相对 PEM 和 S/MIME 的严格基于层次的认证机制的安全电子邮件协议,MOSS 协议具有下列优越性:①使用 MOSS 协议时,用户仅需要密钥对(公钥/私钥),不需要拥有个人证书;②MOSS 对公钥的标识进行了扩展,可以使用任意字符、电子邮件地址或惟一的名字来标识公钥;③MOSS 对算法没有特别的要求,可以使用很多不同的算法。适合于重要机密用户和普通用户混存的机构或单位使用。

2 安全 Web 电子邮件客户端系统结构

本系统的设计,力求与后台邮件服务器软件的功能相配合,为用户提供一个简洁使用的邮件管理接口。严格按照软件开发过程得以实现,即需求分析、总体设计和详细设计的顺序。该系统的每个功能模块都按页面代码 -> 后台代码、用户表示层 -> 业务逻辑层 -> 数据访问层的架构思想得以实现,并采用了组件技术,使整个系统结合的更加紧密、性能更高。

系统的总体结构如图 1 所示。

系统的主要功能如下:

- 具有标准邮件客户端的基本功能:发送邮件、接收邮件、回复邮件、转发邮件等。
- 邮件加解密:确保邮件的机密性,防止邮件被窃听。
- 邮件签名:防止邮件仿冒、邮件被篡改、发方抵赖等。
- 密钥管理灵活,安全性好。
- 支持 GB2312 和 Big5 的内码转换功能。
- 方便易用的地址薄。

限于篇幅,本文简要介绍部分关键模块及使用的关键技术。

3 安全 Web 电子邮件客户端若干关键技术及实现

3.1 MOSS 协议简要分析

MOSS 协议使用两种 MIME 类型(Multipart/Signed 和 Multipart/Encrypted)对 MIME 对象进行封装。MOSS 中使用的 MIME 类型如表 1 所示。

MOSS 协议的安全主要取决于选用的密码算法。我们设计实现的安全 Web 电子邮件客户端采用主流的、安全性好的混合密码算法。摘要算法采用 SHA-1

① 基金项目:重庆市教委科学技术研究项目(编号 KJ050508)

算法，公钥算法采用 1024 位 RSA 算法，对称密码算法采用 21 世纪数据加密标准 AES 算法、分组长度 128 位、密钥长度根据安全性的强弱分别使用 128/192/256 位。

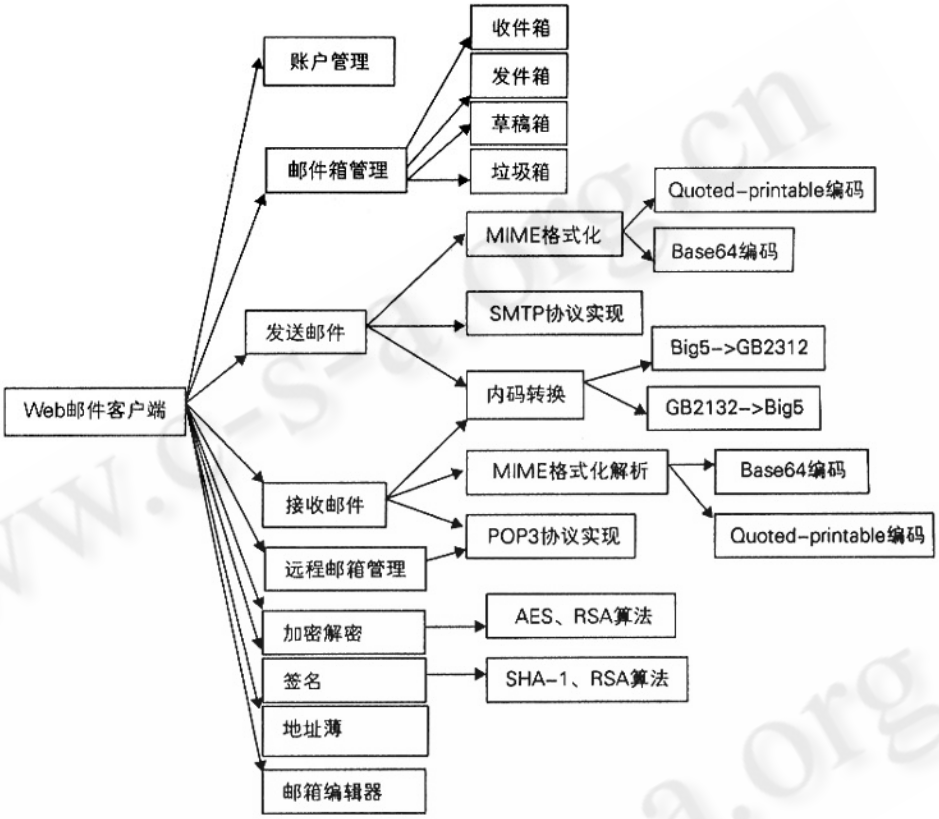

图 1　安全 Web 电子邮件客户端的系统结构

表 1　MOSS 中使用的 MIME 类型

Type	subtype	备注
Multipart	Signed	数字签名封装
	encrypted	加密信息封装
Application	Moss－signature	数字签名
	Moss－keys	会话密钥
	mosskey－request	请求密钥
	Moss－data	发送密钥

MOSS 协议提供了以下功能：①数字签名：防止邮件仿冒、篡改和发送方抵赖；②数据加密：确保邮件的保密性；③密钥管理：公钥、证书验证请求及其应答。

3.2　发送邮件

发送邮件的总过程如图 2 所示。

根据图 2，发送邮件功能模块划分为：撰写邮件（邮件编辑器）、附件添加和删除、格式化邮件（普通邮件、安全邮件）、SMTP 协议实现。

3.2.1　附件添加/删除

附件添加功能实现使用浏览按钮选择本机文件，然后上传到服务器；如果选择多个附件，则将上传的所有附件文件名用逗号隔开。上传的所有附件保存在信息传递目录下 file 中，上传附件后提供给调用页面的数据为上传文件的原文件名和上传后的文件名。

① 通过 get 访问器读取用户控件设置的所有上传附件文件名，通过 set 访问器设置此用户控件的上传附件的文件名。

```
Public string Accessory
{
  get { return txtAccessory.Text; }
```

```
    set { txtAccessory. Text = value; }
  }
```

② 通过 get 访问器读取用户控件设置的所有上传后附件文件名，保存在页面的隐藏控件中，通过 set 访问器设置附件下拉列表。

```
  Public string AccessoryID
  {
```

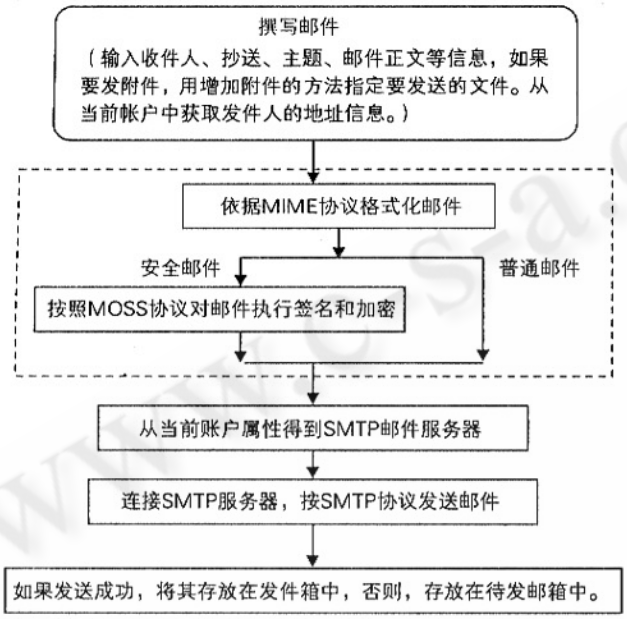

图 2　邮件的发送过程

```
    get { //txt AccessoryID 文本控件为隐藏控件，用于保存附件 ID
      return txtAccessoryID. Text;
    }
    set {
      txtAccessoryID. Text = value;
      char [ ] seprator = new char[1];
      seprator[0] = ',';
      string [ ]acid = txtAccessoryID. Text. Split ( seprator,5 );
      for( int i = 0;i < acc. Length;i + + )
        dopAccessory. Items. Add ( new ListItem ( acc[i],acid[i] ) );
    }
  }
```

③ 邮件“发送”按钮单击方法，首先将上传文件的文件名经过转换后添加到上传附件文本控件中，再将上传附件添加到上传附件下拉列表中，最后将文件上传到指定的目录下。（代码略）

④ 附件“删除”按钮单击事件方法，将附件下拉列表中选中的附件删除，并同步将附件文本控件中的对应文本删除，实现的源代码略。

3.2.2 格式化邮件

本系统严格按照 MIME 协议格式化邮件，下面简述安全邮件的格式化过程。

(1) 对邮件签名：数字签名的过程如图 3 所示。

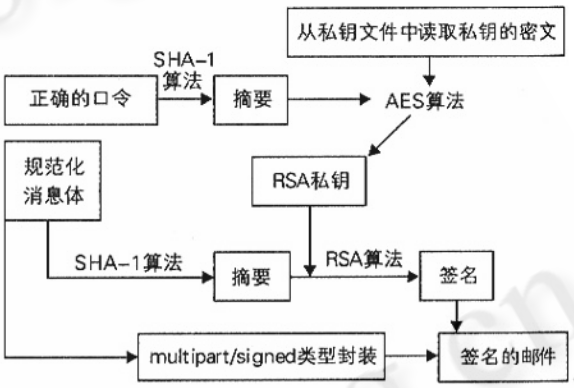

图 3　邮件数字签名过程

图 3 中每一步骤的具体实现如下：

① 规范化要签名的消息体。为了保证发送者和接收者用于计算数字签名的数据是一致的，必须要将用于执行签名的数据规范化。首先对消息体进行 quoted – printable 编码或 base64 编码；接着处理消息体中所有的换行符为 <CR> <LF>。

② 产生签名和相关的控制信息。对第①步中规范化后的消息体执行 SHA –1 算法计算其摘要，再使用发送者的 RSA 私钥对摘要执行 RSA 算法得到数字签名并产生相应控制信息；RSA 私钥的获取是利用 AES 算法解密得到的，其输入是用户口令的散列值和对应私钥的密文。签名的控制信息如表 2 所示。

③ 封装控制信息和签名。对第②步得到的数字签名的值进行 quoted – printable 编码，然后使用 MIME 类型 application/moss – signature 封装控制信息和签名。

表 2　签名服务的控制信息

控制信息	说明
Version	MOSS 协议的版本号，目前为 5
Originator – ID	验证签名的公钥标识及产生签名的用户
MIC – Info	数字签名的值，包括哈希算法标识、签名算法标识和实际签名这 3 个参数，参数之间用逗号分隔开。

④ 将签名数据和消息体封装到 multipart/signed 类型中。multipart/signed 类型包括两个部分，第 1 部分为用于签名的消息，第 2 部分为具体的签名内容。它带有 3 个参数，分别为 boundary、protocol 和 micalg。封装时，首先将参数 protocol 的值设置为 application/moss – signature，将参数 micalg 的值设置为第③步中哈希算法的标识（如 rsa – SHA1），产生签名边界字符串并赋予 boundary 参数；接着对用来产生签名的消息体进行适当的编码，放入第 1 部分，并设置相应的内容类型；最后将控制信息和签名放入第 2 部分，设置内容类型为 application/moss – signature。

（2）对邮件加密。加密采用对称加密算法与公钥算法相结合的方法。用一个随机数生成会话密钥（每次加密不同）采用 AES 算法对明文加密，然后用收方的公钥采用 RSA 对该密钥加密。数据加密过程如图 4 所示，加密后的邮件即为一封安全电子邮件。

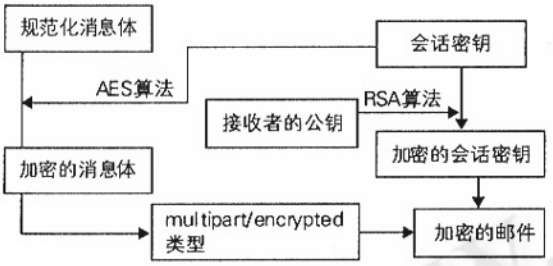

图 4　邮件加密过程

图 4 中每一步骤的具体实现如下：

① 规范化要加密的消息体

同数字签名服务一样，消息体必须转换成尽可能多的系统都能识别的格式。

② 产生会话密钥和控制信息

发送方利用随机生成器随机产生会话密钥，并使用该密钥对第①步中得到的消息体执行 AES 加密，再使用接收者的公钥对该会话密钥执行 RSA 算法并产生相应的控制信息。

加密的控制信息如表 3 所示。

表 3　加密服务的控制信息

控制信息	说明
Version	MOSS 协议的版本号，目前为 5
DEK – Info	用于加密数据的算法和模式
Recipient – ID	加密会话密钥的公钥标识及加密的用户
Key – Info	用接收者的公钥加密后的会话密钥，包括加密算法标识和实际加密密钥两个参数，参数之间用逗号分隔开

Recipient – ID 和 Key – Info 这两个控制信息通常成对出现。当有多个收件人时，通过相应的 Recipient – ID 信息，每个收件人都能正确地解密会话密钥。

③ 封装加密会话密钥和控制信息

对第②步得到的加密会话密钥进行 quoted – printable 编码，然后使用 MIME 类型 application/moss – keys 封装密钥和相应的控制信息。

④ 将加密数据和加密会话密钥封装到 multipart/encrypted 类型中

multipart/encrypted 类型包括两个部分，第 1 部分为加密的会话密钥，第 2 部分为使用会话密钥加密的消息体。它带有两个参数，分别为 boundary 和 protocol。封装时，首先将参数 protocol 的值设置为 application/moss – keys，产生加密边界字符串并赋予 boundary 参数；接着将第（3）步中得到的数据和控制信息放入第 1 部分，设置内容类型为 application/moss – keys；最后对加密数据进行编码，放入第 2 部分，设置内容类型为 application/octet – stream。

3.3　接收邮件

接收邮件的总过程如图 5 所示。

根据图 5，邮件接收分为两个模块：POP3 协议的实现和解析邮件，具体实现略。

4　结束语

本文叙述了安全 Web 电子邮件客户端系统的总体设计方案及具体实现中的若干关键技术。我们在.NET 环境下用 ASP. NET、C# 语言等技术设计实现了基于 MOSS 协议的安全 Web 电子邮件客户端软件，既具有标准电子邮件客户端的功能，又增强邮件的保密性、完整性、身份认证和不可抵赖性等安全功能。该系统经过一年多的时间运行检验，效果较好。

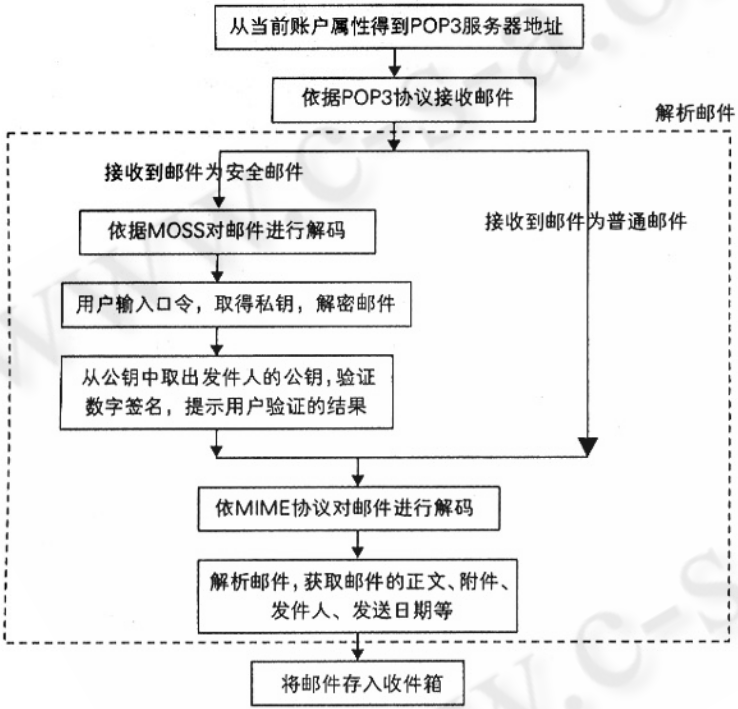

图 5　邮件的接收过程

参考文献

1 S. Crocker, N. Freed, J. Galvin, et al. MIME Object Security Services [S]. RFC 1848. October 1995.

2 J. Galvin, S. Murphy, S. Crocker, et al. Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted [S]. RFC 1847. October 1995.

3 许晓东、荆继武，安全电子邮件的系统分析与密钥管理[J]，计算机应用研究，1999，16(11)：59－61。

4 陈建奇、张玉清、李学农等，安全电子邮件的研究与实现[J]，计算机工程，2002，28(6)：121－122。

5 卿斯汉，密码学与计算机网络安全[M]，北京 清华大学出版社/广西 科学技术出版社，2001.7，236－242。