一种 Web 服务器间的安全传输机制设计与实现①

Design and implementation of Safe Transmission Mechanism Between Web Servers

霍 英 (广东韶关学院 韶关 512005、 中南大学 长沙 410008)
鲁向前 丘志敏 (广东韶关学院 韶关 512005)

摘要:针对一种复杂的多层分布式网络应用系统,本文提出了一种在上下两层 Web 服务器间实现由 ASP 程序调用的透明的、安全的传输机制,并给出了一个实现概要。

关键词:分布式 Web 服务器 传输机制

1 引言

计算机技术结合通讯技术所产生的网络应用发展非常迅速,多层次分布式网络应用模型越来越得到用户的肯定,这种系统适合于分层次管理,功能配置灵活,符合大多数应用环境的实际需求情况。

对于这样一种复杂的应用系统设计,涉及的领域很多,而网络信息的传输安全是网络应用的基本保证。本文以基于 Internet 的多层次分布式网络考试系统为研究背景,从系统信息的传输安全角度进行了研究,提出了一种在上下两层 Web 服务器间实现由 ASP 程序调用的透明的、安全的传输机制。

2 系统总体架构及设计概要

本文研究的基于 Internet 的多层次分布式网络考试系统由一个考试中心和若干个考务中心和考点组成:考试中心下设若干考务中心,每个考务中心下分设若干考点。考试中心负责考试基本信息管理、考题发布、考试系统参数设置、试题回收及相关管理等;考务中心负责考试数据的生成、审核、上报及相关管理等;考点系统负责组织实施具体的考试及相关管理。

为了便于阐述,把系统应用环境抽象为三层次分布应用模型:核心管理层、应用服务管理层和客户服务层,这并不影响安全传输研究的代表性。从系统模型上看,网络传输有两种类型,一是各类用户的浏览器(Browser)与其对应的 Web 服务器(Web Server)之间的信息传输(B – Web 传输),另一类是各 Web 服务器之间的传输(Web – Web 传输)。

B – Web 之间的安全传输是指在浏览器和 Web 服务器两者之间确保网页传输、消息传输、消息响应、表单提交、控件下载等会话事务的安全; Web – Web 之间的安全传输是指在上下级 Web 服务器之间提供安全的文件下载、数据回收、参数设置等批量数据传输。因此这两类传输所需的安全实现是有差别的。

对于 B – Web 之间的安全,可以采用目前因特网上 Web 服务器实现安全的事实标准 SSL,关于这方面的讨论有很多,本文对此不作研究;而对于 Web – Web 之间的安全,由于系统要求是在两层 Web 服务器内核实现透明的、安全的传输,其传输数据又具有批量性的特征,故本文提出一种在应用层实现、由 ASP 驱动、基于文件安全传输的解决方案。本解决方案包含两方面内容:一是通过加密来实现数据的安全,对数据安全的各种属性进行了针对性的分析,并给出了一个详细的加密流程图;二是对传输方案进行了研究,提出了这样一种方案:上层 Web 服务器中内置一个 NT 服务器程序,下层 Web 服务器中内置一个 DLL 形式的客户机程序,并各自绑定安全功能,然后由用户驱动 ASP 程序去调用它以实现安全传输。本文 3、4 节将详细阐述这两方面内容。

① 基金资助情况:本课题得到国家自然科学基金(60573127);国家教育部博士点基金(20040533036)资助

3　Web－Web 间传输数据的加密算法研究

网络传输中安全的实现通常是对传输数据进行加密，目前密码编码算法分三大类：对称密码算法、公开密钥算法、消息摘要算法。要真正自己去实现这些算法以及相应的一些协议，工作量是非常巨大的。因此作为实现，我们应该采用现存的一些密码算法软件包来实现二次开发。

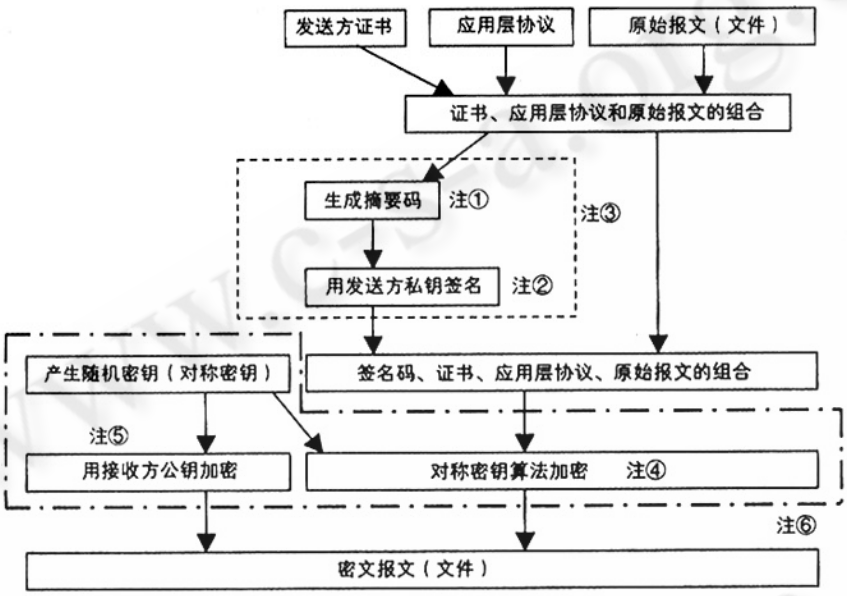

图 1　加密流程图

加拿大人 Eric A. Young 和 Tim J. Hudson 开发的 OpenSSL 包，它开放源码，支持 SSL3.0 和 TLS，C 语言开发，可跨平台。它目前的最新版本是 0.9.7d 版。OpenSSL 支持 Linux、Windows、BSD、Mac、VMS 等平台，这使得 OpenSSL 具有广泛的适用性。OpenSSL 的算法目录 Crypto 目录包含了 OpenSSL 密码算法库的所有源代码文件，是 OpenSSL 中最重要的目录之一。OpenSSL 的密码算法库包含了 OpenSSL 中所有密码算法、密钥管理和证书管理相关标准的实现，在 Windows 下编译后生成两个文件 libeay32.lib、libeay32.dll，这两个文件是本系统设计安全传输的重要基础。

加密算法主要针对传输中的机密性、完整性、不可抵赖性而设计，加密设计如图 1 所示。

• 注①所示步骤（即生成摘要码）是为了保证数据的完整性，在 OpenSSL 中用 EVP_Digest 系列函数实现；

• 注②所示步骤（即用发送方私钥签名）是为了实现发送方的不可抵赖，在 OpenSSL 中用 EVP_ private _decrypt 函数实现；

• 注③所示步骤（即包含注①和注②）是指在 OpenSSL 中可以用 EVP_Sign 系列函数集成实现，即在 EVP_Sign 系列函数中包含了生成摘要码和私钥签名的功能实现，为我们的设计提供了方便。很多时候我们是把这个合成步骤统称为数字签名，很容易与私钥签名的概念混淆；

• 注④所示步骤（即对称密钥算法加密）是为了保证数据的机密性，在 OpenSSL 中用 EVP_Encrypt 接口实现，由于三级 Web 服务器都是在同一系统内，因此为了实现的方便性，可以采用统一的加密算法，比如都使用 3DES 算法，但如果要与外部系统交换数据，可以使用统一接口实现方便的转换；

• 注⑤所示步骤（即用接收方公钥加密）在标准的 SSL 实现中是为了交换对话密钥，实现一次会话过程的多次报文的往返传输安全，因此它在实现上是把交换对话密钥与报文加密分成两个步骤的，但是在 Web－Web 之间的传输类型不是以会话为主，而是以批量数据的传输为主，因此在实现上可以把它与传输报文绑定，这样可以简化传输过程。在 OpenSSL 中用 RSA_public_encrypt 函数实现；

• 注⑥所示步骤（即包含注④和注⑤）是指在 OpenSSL 中可以用 EVP_Seal 系列函数集成实现，即在 EVP_Seal 系列函数中包含了公钥加密和对称加密两个步骤的功能实现，为我们的设计提供了方便。

加密过程用算法表示如下：

```
bool Sf_To_Df(void &DestPackage, char * ScertFile,
char * SDatabase, char * STable, char * DHost, char
```

```
* DDatabase, char * DTable, char * OperateType)
{
  生成应用层协议;
  组合证书、应用层协议和源报文 =》NewPackage1;
  EVP_Sign(NewPackage1, Private_Key) =》NewPackage2;
  EVP_Seal(NewPackage2, Public_Cert) =》DestPackage;
}
```

4 Web－Web 间传输方案研究

一般情况下，网络中数据传输有以下几种方案：

（1）使用标准的 Windows2000 Server IIS 提供的 FTP 服务器和标准的浏览器实现传输。

（2）使用标准的 Windows2000 Server IIS 提供的 FTP 服务器和自己设计的客户端程序实现传输。

（3）使用自己设计的 FTP 服务器程序和标准的浏览器作为客户端实现传输。

（4）使用自己设计的 FTP 服务器程序和自己设计的客户端程序实现传输。

前三种传输方案在实现上相对简单，但并不能满足在 Web－Web 间实现透明的、安全的传输需求，本文选择第四种方案。但是我们没有必要去实现一个全功能的 FTP 传输再外加扩展的安全功能，因为标准的 FTP 协议包含了许多像更改目录等我们在工程中不必使用的部份，因此我们可以根据工程实际情况简化 FTP 协议，然后再外加工程中需要扩展的功能。

按照技术发展的先后，我们可以从四个层次进行网络应用程序的开发：① 使用 Sockets API；② 使用 Windows Socket 类；③ 使用 WinInet API；④ 使用 WinInet 类。前两种方式既可以用来编写服务器程序，也可以用来编写客户机程序；而后两种方式只能用来编写客户机程序，而且必须是诸如 HTTP、FTP 这类已经有了标准协议的客户机程序，即用后两种方式编写的客户机程序只能用于访问使用标准协议的服务器程序。

经过分析和验证，我们可以使用 WinInet 类来编写客户机程序，而使用 Socket API 来编写一个简化了的标准的 FTP 协议的服务器程序，并且在服务器程序和客户机程序中都绑定安全功能和数据库访问功能。

FTP 协议使用简单的命令/应答方式传送文件。它使用两条单独的 TCP 连接，一条是用于专门传送双方的命令和应答的控制连接，另一条才是真正用于传送数据的数据连接。控制连接使用 21 号端口，而数据连接使用 20 号端口。服务器程序先在 21 号端口监听客户机的连接请求；客户机需要传送文件时可以向服务器的 21 号端口发送连接请求。控制连接建立后，客户机程序可以向服务器程序发送文件下载或上传的请求（还是通过 21 号端口发送），并且客户机程序还要在 20 号端口建立一个监听套接字，以使服务器程序能够反向向客户机程序发送数据连接请求；而服务器程序在收到客户机程序发送来的文件下载或上传命令后，要向客户机程序的 20 号端口发送一个反向的数据连接请求。当数据连接建立好后，双方可以开始文件传输。

5 Web－Web 间安全传输方案的实现

对于 Web－Web 之间的安全传输，需要实现三个模块，分别是：

- 下级 Web 的后台文件传输客户端程序（内置加密/解密和数据库访问功能）
- 下级 Web 的 ASP 程序
- 上级 Web 的后台文件服务器程序（内置加密/解密和数据库访问功能）

（1）下级 Web 的后台文件传输客户端实现：为了实现用户的核心商业秘密和软件流程的秘密，把商务逻辑程序写成 DLL 后由 ASP 调用。实现 DLL 有多种方法，为了实现面向对象的程序设计方法，可使用 COM 技术。建立一个 InetAPP. DLL 文件，包含一个对象 RMD_Trans，该对象包含两个方法：Get_File( ) 和 Put_File( )。下面给出 Get_File 方法的实现逻辑，Put_File 方法类似，在此略去。

```
STDMETHODIMP CRMD_Trans::Get_File(BSTR * siteName, BSTR * str1, BSTR * str2, BSTR * retMsg)
{
  ……
  //以下代码是用 WinInet 类实现文件传输的客户端程序
  CInternetSession * pInetSesn = new CInternetSession();//生成对象
```

```
  ……
  if( NULL = = ( pConn = pInetSesn - > GetFtpConnection( siteStr, UserName, Password, 21) ) ){
     * retMsg = str3; //连接服务器失败
    return S_OK; }
  if( NULL = = pConn - > GetFile ( RemoteFile, LocalFile) ) {
     * retMsg = str4; //下载文件失败
    delete pConn;
    return S_OK; }
  pInetSesn - > Close( );
  bool b = Df_To_Sf( Df, Sf) ; //把 Df 解密成 Sf。
  ……
}
```

(2) 下级 Web 的 ASP 程序实现:包含两个 ASP 程序:Put_File. asp 和 Get_File. asp。Get_File. asp 代码如下:(Put_File. asp 程序类似,在此略去)

```
<%
  Dim obj123
  Set obj123 = Server. CreateObject ( " InetAPP. RMD_Trans" ) //生成对象
//连接上层服务器,下载文件并进行解密
Response. Write obj123. Get _ File ( IpAddress, UserName, Password)
  Set obj123 = Nothing //关闭对象
% >
```

(3) 上级 Web 的后台文件服务器程序实现:服务器程序必须能够并发处理多个客户机的服务请求,Windows 的多任务调度技术使得服务器可以给每个客户机请求创建一个线程,独立的处理请求和应答,一个处理线程的阻塞不会妨碍其它线程的服务正常进行,因此是开发 Windows 服务器程序的理想方法。Windows 有一个重要的系统程序——服务控制管理程序,系统中由它管理所有基于 NT 技术的服务器程序,比如我们常见的 IIS,FTP 等典型的服务程序都是由服务控制管理程序进行控制。服务控制管理程序提供了统一的和安全的接口,使得程序员只要遵照接口标准就可以方便地编写出自己的服务器程序。关于 NT 服务器程序的接口标准,由于很多资料都有介绍,在此只给出协议解释线程函数中的实现逻辑:

```
unsigned_stdcall FtpPIThread( void * pArg)
{
  连接请求的用户身份合性检测;
  while( true)
  {……
  如果命令动词是 STOR://上传文件的请求
      {
      接收数据;
      解密;
      }
    如果命令动词是 RETR: //下载文件的请求
     {
      加密;
      把数据发送到请求方;
     }
  }
}
```

6 结束语

本文对于 Web 服务器间安全传输机制的研究和实现,提出了一套在 Web - Web 之间进行透明的安全的传输方案,它为分布式协作服务器的应用提供了一个非常有意义的思想,为构建大型安全网络提供了一个好的参考方案。同时它还为 Web 服务器的安全防范提供了一个有意义的参考。

在今后的工作中,对于 Web 服务器间实现传输的内部逻辑流程及传输的灵活性需求,还有改进和提高的研究空间。

参考文献

1 邵兵、鲁东明,一个数据加密传输系统的研究与实现[J],计算机应用,2000,20(6)。

2 吴凯、陈晓苏、肖道举,网络传输层安全协议 SSL 的安全研究[J],计算机系统应用,2003.1。

3 葛丽娜、钟诚、石润华,网上考试系统的一种身份认证方案[J],微机发展,2003, 13(9)。

4 William Stallins,密码编码学与网络安全:原理与实践(第二版)[M],北京 电子工业出版社,2001。