第 20 卷 第 3 期 华侨大学学报(自然科学版) Vol. 20 No. 3
1999 年 7 月 Journal of Huaqiao University (Natural Science) Jul. 1999

利用 JAVA 编程实现网络安全通信*

吴金龙 吕吉实

(华侨大学计算机科学系, 泉州 362011)

摘要 网络安全涉及面极广, 而数据传输的准确性、机密数据的安全性是计算机网络安全通信的关键. 在 INTERNET 网络环境下, 讨论利用 JAVA 语言实现网络安全通信的编程特性. 着重说明基于 TCP/IP 协议的 JAVA. Net. Socket 类库的通信机制, 以及客户机/服务器工作模式. 给出实现 JJCP 系统的文件传送流程图和多线程结构图, 列出用户服务线程的部分源代码. 同时, 针对数据加密技术进行较详细的分析.

关键词 网络安全, 数据加密, JAVA 编程
分类号 TP 393: TP 311.5

随着网络技术的发展和 INTERNET 的广泛应用, 计算机网络的安全问题正受到国际社会的普遍关注. 网络安全涉及面极广, 其中数据传输的准确性、机密数据的安全性则是众多问题中的关键, 也是实现网络资源共享的前提. 由于网络环境的复杂性, 不同的硬件平台、不同的操作系统、不同的图形用户界面 GUI 等既有连接, 又不兼容. 这样, 就促使网络用户开发一种新型的跨平台分布式程序设计语言(JAVA 语言). 本文主要论述在 INTERNET 环境下, 利用 JAVA 语言实现网络安全通信的编程特性, 并对数据加密技术进行较详细的分析.

1 JAVA 与 INTERNET

通过公共网关接口 CGI, 用户就可以利用 WWW 进行网上通信和贸易等活动, 实现 HTTP 协议所不能实现的 Internet 服务. 但是, CGI 的程序功能是完全在服务器上实现的, 且受传输速度和带宽的限制, 难于实时响应. 要解决这一问题的关键, 是在客户机/服务器模式中加大客户机一方的工作量. 同时, 为 Internet 寻找一种语言, 编写在客户端执行的程序, 当客户机提出请求时发送给客户机. 在 Interent 上传送程序并不难, 问题在于传送的程序如针对某一平台编写或编译的, 就不可能运行于所有平台上. JAVA 的运行机制, JAVA 源代码→JAVA 字节码→解释执行的过程, 正好解决 WWW 期待已久的问题. JAVA 源代码经过编译后, 生成独立于平台的字节码放在服务器上. 当客户机提出请求时, 从客户机上下载相应的字节码到本地, 由本地的 JAVA 解释器解释执行 JAVA 字节码程序. 因此, JAVA 成为一种真正的跨平台设计语言. 除此之外, JAVA 是一种实际意义上的分布式的面向对象的语言. 它在强键性、安全性和高效率方面也有良好的表现, JAVA 的多线程机制使网络用户受益匪

* 本文 1999-01-20 收到; 福建省自然科学基金资助项目

浅[1]. 只要继承 Thread 这个基本类, 可以利用原编写的方法, 生成一个新的线程, 执行一个线程, 终止一个线程, 或者查看执行状态.

2　网络安全与 JAVA 编程特性

JAVA 是针对网络环境的程序设计语言, 以完全面向对象的类形式提供两个不同层次的网络支持机制. 其一, 是利用统一资源定位符URL, 访问网络资源的应用类库. 使用这些类时, 用户不需要考虑 URL 中标识的条件协议的处理过程, 就能直接获取 URL 资源信息. 特别是, 这些类库对超文本传输协议 HTTP 提供了更广泛的支持. 其二, 是利用 Socket 实现 CLIENT/SERVER 通信模式的类库. 基于 TCP/IP 协议的 JAVA.Net.Socket 类库支持两种通信方式. (1) 有连接的流方式. 在该连接上, 以同一进程实现可靠的、有序的、无差错的、分组长度不定的、全双工的、不重复的字节流服务. (2) 无连接的数据报方式. 它不要求在初始化时建立连接. 每次网络 I/O 操作可以在不同主机、不同进程之间进行, 以最大长度为 32 KB 的独立分组进行传输. 但其传输不保证顺序性、可靠性、无重复性, 也不保证数据报一定到达目的地. 显然, 流方式较数据报方式可靠、安全. 我们利用 JAVA 的编程特性, 以客户机/服务器模式, 实现 Java to Java Copy 的文件传送, 其程序流程图如图 1 所示.

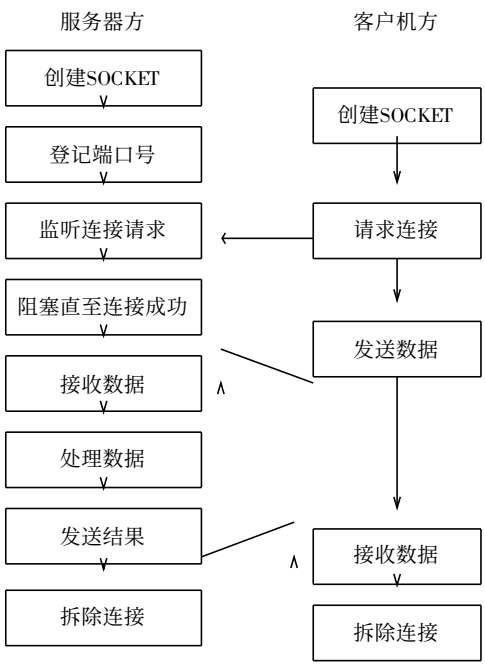

图1　JJCP程序流程图

在客户机/服务器之间所连接的通信组件中, 一般由一个地址加上一个端口号来标识. 每个地址上最多可存 65 535 个端口, 每一个服务程序都在一个端口上提供服务. 例如, HTTP 的端口号为 80, FTP 的端口号为 21, TELNET 的端口号为 23 等, 要使用某个服务的客户机程序就需要与该端口连接. 基于 TCP/IP 协议的网络系统一般为公共服务保留 0～1 024 作为连接端口号, 而 1 024～65 535 之间的端口, 可供程序员选用. 我们编写的 JJCP 程序就是利用 4 466 作为主要的命令端口, 当进行数据传送时, 动态地占用 4 467 作为数据传送端口.

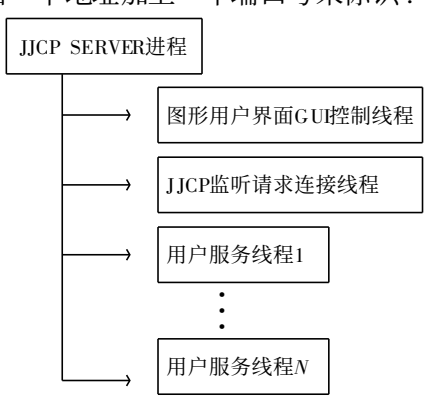

图2　JJCP多线程结构

JJCP 的多线程结构如图 2 所示. 利用 JAVA 的多线程机制, 是我们设计在 JJCP SERVER 程序中的又一个特性. 在一个进程中, 利用第一个线程来执行图形用户界面, 它提供图形刷新、文件列表

刷新和鼠标活动等控制功能．第二个线程被用于监测用户连接活动,当有用户请求建立连接时,该线程就监测其活动,阻塞其余线程,直到连接成功．第三个线程是第一个用户连接成功后产生的用户服务线程．第四个线程将是下一个有效用户连接生成的用户服务线程．每个用户对应一个 Connection 服务线程,以便实现多用户共享文件．

下面列出 JJCP 程序中监听请求连接线程和接收连接请求,并生成 Connection 用户服务线程的源代码．程序中设置的缺省命令端口号为 4 466.

```
public class JJCPServer extends Thread{
  public final static int DEFAULT -
      PORT= 4466;
  protected int port;
  protected ServerSocket listen - socket;
  Connection connect;
  Server gServer;
  //create a listen tread.
  public JJCPServer(int port, Server ser){
    this. gServer= ser;
    if(port= = 0)port= DEFAULT -
        PORT;
    this. port= port;
      try{
          listen-socket= new
          ServerSocket(port);
      }
      catch(IOException e){
        System. err. println( "Exception
          creating server socket ");
      }
      System. err. println( "Server: Local
        port is " + listen-socket.
        getLocalPort());
      System. err. println( "Server:
        listening
        on port " + port);
        this. start();
  }
  //server running body.
  public void run(){
    try{
      while(true){
        Socket client - socket= listen -
          socket. accept();
        connect= new Connection
        (client - socket, gServer);
        gServer. newUser(connect);
        try{
            sleep(5);
        }catch(interruptedException e){;
              }
        }
    }
    catch(IOException e){
        System. err. println( "Exception
          while
          listening for connections ");
        }
  }
  public Connection getConnection(){
        return this. connect;
    }
  }
```

Internet 是一种分布式计算环境,要求运行的软件具有高度的稳定性和安全性．JAVA 语言在这方面具有特别的优势．在稳定性方面,JAVA 采取 3 项措施．(a) 不支持指针数据类型,避免通过指针访问任意内存．(b) 提供数组下标检查机制．(c) 提供内存碎片收集器．在安全性方面,除了在字节码的传输过程使用公开密钥外,JAVA 的运行环境提供了 4 级安全保障机制．(1) 字节码校验器;(2) 类装载器;(3) 运行时间内存布局;(4) 文件访问限制[3].

JAVA 网络软件包提供处理各种网络协议(如 FIP, HTTP, Telnet 等) 的用户接口, 使用户设置网络访问权限带来了极大的方便 . 此外, JAVA 在运行时间系统中内置了防病毒和文件系统保护机制, 使从网上卸载或修改的 JAVA 应用程序具有安全保障 . 利用 JAVA 语言编程, 加上数据加密技术, 就能较好地实现当前的网络安全措施 .

3　通信安全与数据加密技术

网络资源是一种宝贵财富, 只有实现共享才能显示它应有的价值 . 共享资源又必须以安全通信为前提; 而数据加密技术则是安全通信的重要手段之一 . 数据加密型网络安全技术是通过网络中传输的信息进行数据加密为基础的, 是一种主动型安全防御策略 .

目前, 常用的 3 种数据加密算法. (1) DES 算法也叫对称加密法, 是一种公开加密算法 . (2) RSA 算法也称为非对称密钥加密法, 这是一种公开密钥的公钥密码体制 . (3) 非公开加密算法也叫做传统加密法, 其算法有多种多样, 强度各不相同 . 这 3 种加密算法各有其特点 . 与 DES 相比, RAS 更适合于分布式网络环境 . 由于使用公开的密钥加密, RAS 可极大地简化密钥管理工作的难度, 避免了因密钥交换可能产生的失密问题 . 但是, RAS 的运算复杂, 用户较难确定所选择的 p 和 q 是两个大的安全的素数, 而安全素数本身的概念也不清晰, 实验应用有一定局限性[3]. 因此, 在实际应用中, 常利用对称密钥加密技术运算速度快的特点对信息进行加密 . 然后, 用非对称密钥加密技术中公用密钥的特点, 对前者需要交换的密钥进行加密, 解决密钥交换中的难题 . 几种方法综合使用, 取长补短 .

JAVA 语言是一种安全性语言, 但不等于说利用 JAVA 编程就能自动抵御意图不轨的程序攻击 . JAVA 语言并不保证数据在网络通信中的正确传输, 也不保证数据在存放中的安全保密 . 因此, 数据通信的安全主要取决于数据的加密技术和数据传输协议的正确应用 . 数据的安全性不会因中间节点不可靠而受影响 . 根据加密操作的特点, 加密机制的最佳位置应设在表示层 . 表示层只考虑数据的传输语法, 而不关心数据的语义 . 就 Internet 的网络协议层次来说, IP 协议处在网络层, TCP 协议属于传输层, TCP/IP 协议的应用层提供了 FTP, SMTP, DNS 和 Telnet 等各种应用服务 . 因此, 我们利用 JAVA 语言编写的 JJCP 传输程序是面向应用层的, 其加密机制是设在表示层上, 它不影响正确的路由选择, 也不影响其它网络特性 .

通常情况下, 密码体制应满足几点基本要求. (1) 易于使用. (2) 加密解密变换迅速有效. (3) 安全性只依赖于密钥的保密性, 而不依赖于加密算法或解密算法的保密性 . 据此, 我们经过筛选, 选择 5 种算法用于 JJCP 系统设计: (a) DES 算法模拟; (b) CDED 加密法; (c) 软件黑盒子加密法; (d) 序列加密法; (e) 码变换加密法 . 这些加密算法满足简单、实用、高效的特点. 例如, DES 算法模拟的设计思想是: 先将 32 位种子密钥 K 按某种简单算法(如移位、换位) 产生 K_1, K_2, K_3, K_4; 再将 32 位明文分为左、右两部分 L_0, R_0, 作适当异或、模 256 的加法及移位、交换等运算 . 最后 L_8, R_8 就是 L_0, R_0 的密文 . 密钥使用由 $K_1 \rightarrow K_4 \rightarrow K_1$, 具有可逆性 . 因此, 明文加密成密文, 对密文再运行一次就得明文 . JJCP 系统利用这样的加密技术在 JJCP CLIENT 中实现文件加密系统模块 . 另一个 JJCP CLIENT 通过 JJCP SERVER 获取该文件, 再由自己的 JJCP CLIENT 中的解密系统进行解密 .

通过多次实验表明, JJCP 系统可以实现工作站间的安全对等通信, 文件传送具有良好的

安全实时性. 用户在进行加密操作时,可任意选择其中一种算法进行加密,加密后还能重复选择某种算法再次加密. 攻击者即使获取密文,也只能是乱码,其加密强度可以达到难于破译的地步.

4 结束语

利用JAVA 编程的JJCP 系统,由JJCP CLIENT 及JJCP SERVER 两部分组成. 它要求操作系统为Windows 95/Windows 98/Windows NT 等具有32 位的网络环境. 在只要求在网络属性中加入TCP/IP 协议,并在IP 地址部分输入一个固定的IP 地址以及本机的域名. 当JJCP SERVER 运行后,会显示当前服务器的IP 地址、域名及端口号,并开始接受用户端的录入及监测用户状态. 如果JJCP CLIENT 的程序也运行成功,并在网络上找出给定地址的JJCP SERVER 进行连接,则可获得JJCP SERVER 送来的文件列表. 这样,用户可根据需要对文件进行加密和传递. 反之,对加密的文件可进行解密,只要双方使用正确的密钥,就能实现安全通信. 总之,JJCP 具有良好的操作界面和广阔的应用前景.

参 考 文 献

1 麦中凡,陶 伟,曹广通等. C/C++程序员JAVA 编程. 北京:清华大学出版社,1996. 111～120
2 赖福军. Internet 信息制作与传播技术. 北京:人民邮民出版社,1997. 235～238
3 刘尊全. 刘氏高强度公开加密算法设计原理和装置. 北京:清华大学出版社,1996. 108～122

Realizing Network Safety Communication by Applying JAVA Programming

Wu Jinlong
(Dept. of Comput. Sci., Huaqiao Univ., 362011, Quanzhou)

Abstract The crux of safety communication of computer network lies in the accuracy of data transmission and the security of confidential data. On the realization of network safety communication in the Internet network environment, the authors discuss the characteristics of programming in JAVA laguage. Emphasis is placed on the communication mechanism of JAVA. Net. Socket class library based on TCP/IP protocol; and on file transport and block diagram of multi-threading of JJCP system realizing in client/server model, with a list of parts of source codes of user service threading. A detailed analysis is aimed at the data cryptography.

Keywords network security, data encrypt, JAVA programming