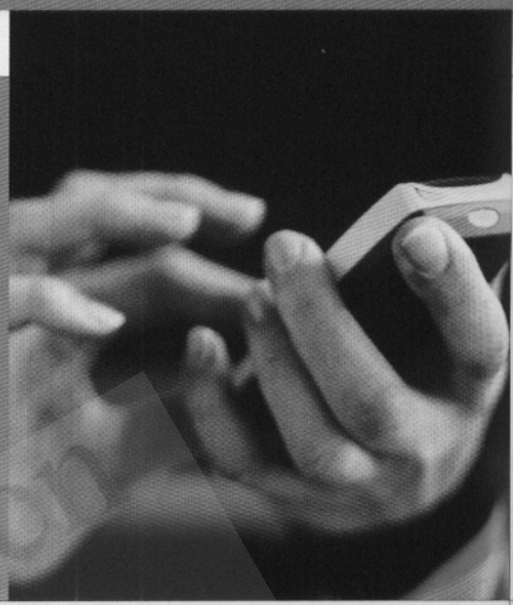
Java Applet 与 Java Servlet 的安全通信策略与实现

The Tactics and Realizing of Security Communication Between Java Applet and Java Servlet



左黎明 刘二根 盛梅波 （南昌华东交通大学基础科学学院 330013）

摘 要：介绍了基于Java技术的三层网络结构，提出了一套基于JCE和MD5消息摘要算法的Applet与Servlet的安全通信策略方案，并详细阐述了该策略的实施过程，实践证明该方案具有很高的可靠性和实用性。

关键词：信息 JAVA MD5 JCE Servlet Applet

1 引言

随着宽带网和高速信息公路的飞速发展，虚拟现实技术和网络实时可视化系统的实现成为可能，网络媒体互动与网络游戏方兴未艾。JAVA技术的不断发展和强大为网络提供了最强有力的技术支持，JAVA技术在各行各业的应用得到了迅速发展，基于Java技术的三层网络结构已经成为网络开发的主流模式。Applet为建立功能强大的动态界面提供了便利的机制，Servlet为web服务器或者其他应用服务器处理请求提供了高效率的手段。如果要开发需要在客户端进行简单操作和图形处理的网络实时系统，那么Applet+Servlet+JDBC+数据库是一种比较理想的模式。同时对于某些网上银行证券系统、档案管理系统、成绩发布系统和股票系统而言，许多数据都比较重要，机密性要求较高，因此安全问题是不容忽视的问题。

2 基于Java技术的三层网络结构

三层网络结构，指的是将数据处理过程分为三部分：第一层是客户端(用户界面层)，提供用户与系统的友好访问；第二层是应用服务层（也叫中间层），专司业务逻辑的实现；第三层是数据服务层（数据库系统），负责数据信息的存储、访问及其优化。由于业务逻辑被提取到应用服务层，大大降低了客户端负担，因此也成为瘦客户(Thin Client)结构，三层结构在传统的二层结构的基础上增加了应用服务层，将应用逻辑单独进行处理，从而使得用户界面与应用逻辑位于不同的平台上，两者之间的通信协议由系统自行定义。通过这样的结构设计，使得应用逻辑被所有用户共享，这是两层结构应用软件与三层应用软件之间最大的区别。三层结构将表示部分和业务逻辑部分按照客户层和应用服务层相分离，客户端和应用服务层、应用服务层和数据库服务层之间的通信、异构平台之间的数据交换等都可以通过中间件或者相关程序来实现。当数据库或者应用服务层的业务逻辑改变时，客户端并不需要改变，反之亦然，大大提高了系统模块的复用性，缩短开发周期，降低维护费用。以Java Applet为客户端，以Java Servlet为中间层的三层网络结构，在目前的实时网络信息平台得到了广泛的应用，其结构如图1所示：

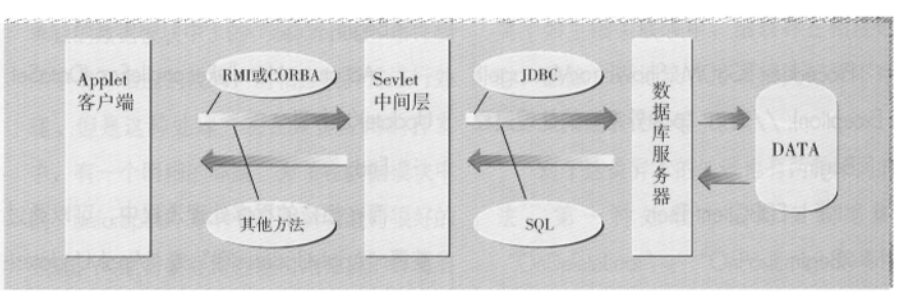


图1 基于Java技术的三层网络结构（Applet为客户端，Sevlet为中间层）

3 Applet与Servlet的安全通信设计方案

3.1 Applet与Servlet的通信方法

Sun公司推荐的一种规范是：在前端使用Applet、HTML和JSP，在后端使用Enterprise JavaBeans支持的Servlet及其他成分。这种体系结构的关键是在客户端的Applet和在服务器端的Servlet之间的通信。但是由于Applet受浏览器安全模式的限制，在一个Applet中存取数据和信息并不想看上去的那么简单，在较为复杂的分布式Java应用中，Applet与Servlet之间的通信方式主要有以下四种：

（1）通过HTML页面传递参数（也称HTTP隧道）；

（2）用java.net包的网络功能建立直接网络连接（也称SOCKET隧道）；

（3）使用远程方法调用（RMI）技术；

（4）使用CORBA技术。

使用RMI、CORBA技术在Applet和Servlet之间建立对话理论上很完善，在实际中很难普及。因为在实际应用中不可能要求所有用户在客户端安装其所有支持文件，同时由于Applet采用的是“沙箱”机制，更加大了其实现难度。因此，第（1）种和第（2）种方法对客户端没有什么特殊的要求，考虑到大多数实际应用的需要，我们应该多采用这两种方法。

3.2 JCE简介

Java加密扩展即Java Cryptography Extension，简称JCE。它是Sun的加密服务软件，包含了加密和密匙生成功能。JCE是JCA（Java Cryptography Architecture）的一种扩展。

JCE没有规定具体的加密算法，但提供了一个框架，加密算法的具体实现可以作为服务提供者加入。除了JCE框架之外，JCE软件包还包含了SunJCE服务提供者，其中包括许多有用的密码算法和数字签名算法，例如DES、RSA、DSA、SHA、MD5等。

3.3 Applet与Servlet的安全通信设计

在Applet与Servlet之间实现安全通信，就是要保证Applet与Servlet之间的信息流未被攻击者修改和通信过程的合法性。要实现这一点，一般有两种做法：

（1）使用消息摘要；

（2）使用数字签名。

MD5是比较成熟和快速的消息摘要和数字签名算法，它将一段数据信息（Message）通过其不可逆的字符串变换，产生一个唯一的MD5信息摘要。如果在以后传播这段数据信息（Message）的过程中，无论数据信息（Message）的内容发生了任何形式的改变（包括人为修改或者下载过程中线路不稳定引起的传输错误等），只要对这段数据信息（Message）重新计算MD5信息摘要时就会发现信息摘要不相同，由此可以确定你得到的只是一个不正确的数据信息（Message）。如果再有一个第三方的认证机构，用MD5还可以防止文件作者的"抵赖"，这就是所谓的数字签名。我们利用Java的JCE技术和HTTP对象流隧道技术，同时采用MD5算法就可以保证大部分应用中的Applet与Servlet之间实现安全通信，其原理步骤如下：

① 端向中间层提出数据请求；

② 中间层接受数据请求，执行相关的数据处理操作；

③ 中间层对要返回的信息产生一个唯一的MD5信息摘要；

④ 利用HTTP对象流返回信息和MD5信息摘要给客户端；

⑤ 客户端得到中间层返回的信息和MD5信息摘要，对返回信息重新计算MD5信息摘要，与中间层发过来的信息摘要进行比较，如果相同，这确认得到的信息正确，做进一步处理，如果不同，则丢弃得到的信息，处理异常后重新转向（1）。关键方法说明和部分代码如下：

·关键数据和方法说明：

引入java.security包：import java.security.*;

中间层需要返回的信息：message

获得MD5实例：MessageDigest middleMD=MessageDigest.getInstance("MD5")

添加要进行计算摘要的信息：middleMD.update(message.getBytes())

计算MD5摘要：byte[] msgdigm= middleMD.digest()

通过HTTP对象流发送给客户端的信息和摘要，客户端用相同的方法初始化，添加信息,最后进行比较摘要是否相同：

clientMD.isEqual(msgdigm,clientMD.digest())

·中间层Servlet关于生成消息摘要的部分代码：

```
import java.security.*;
public String creatDigest(String message)
{
String msgabsM=new String();
try {
MessageDigest middleMD=MessageDigest.getInstance("MD5");//获得MD5实例
middleMD.update(message.getBytes());//添加要进行计算摘要的信息
byte[] msgdigm= middleMD.digest();//计算出MD5摘要
 msgabsM= byte2hex(msgdigm);}
catch (NoSuchAlgorithmException ex)
{
```

```
异常处理代码;
//由于JCE可能出现“非法或不存在算法”异常，需做异常处理
}
return(msgabsM);
}
```

·客户端Applet关于验证消息摘要的部分代码:

```
//客户端收到信息(message)和摘要(msgabsM)，判断信息是否被更改或传输正常
public Boolean creatDigest(String message，String megabsM)
{ Boolean testanswer=new Boolean();//定义一个逻辑变量
byte[] msgdigm=megabsM. getBytes();
try {
MessageDigest clientMD=MessageDigest.getInstance("MD5");
client.update(message.getBytes());
/*下面判断摘要是否相同*/
if (clientMD.isEqual(msgdigm, clientMD.digest()))
{ testanswer=TRUE;
// 信息检查正常}
else
{testanswer=FALSE;
//摘要不相同}
}
catch (NoSuchAlgorithmException ex)
{
异常处理代码;
//由于JCE可能出现“非法摘要算法”异常，需做异常处理
}
return(testanswer);
}
```

·为网络传输方便，将二行制转化为字符串

```
public String byte2hex(byte[] b) {
String hs="";
String stmp="";
for (int n=0;n<b.length;n++)
{
stmp=(java.lang.Integer.toHexString(b[n] & 0XFF));
if (stmp.length()==1) hs=hs+"0"+stmp;
else hs=hs+stmp;
if (n<b.length-1) hs=hs+":";
}
return hs.toUpperCase();
}
}
```

·Applet的Servlet之间的通信（采用HTTP对象流隧道）

```
/*先构造一个MSG对象，包含信息message和MD信息摘要，然后生成一个对象实例msg1，
msg1这个对象必须执行Serializable接口以使其序列化。*/

//中间层Selvlet关键代码
public void doGet(HttpservletRequest req, HttpservletResponse res)
throws servletException, IOException {
  OutputStream out;//创建一个输出流
  ObjectOutputStream objStream;//创建一个对象输出流
  out = res.getOutputStream();
  objStream = new ObjectOutputStream(out);
  out.writeObject(msg1);
}
//客户端Applet关键代码, msg2这个对象必须执行Serializable接口以使其序列化
private MSG getMessage() throws MalformedURLException
IOException {
    URL url = new URL(getCodeBase(), "/servlet/MD5servlet");
    URLConnection con = url.openConnection();
    con.setUseCaches(false);
    InputStream in = con.getInputStream();//创建一个输入流
    ObjectInputStream objStream; //创建一个对象输入流
    objStream = new ObjectInputStream(in);
    MSG msg2 = (MSG)objStream.readObject();
    return msg2;
}
/*最后从对象msg2中分离出信息message和MD信息摘要，接着执行验证操作（代码略）*/
```

4 应用实例与分析

在远程股票信息系统开发中，我们采用了三层结构，开发工具包括：Jbuilder8.0、Weblogic7.0和Tomcat4.0，数据库服务器采用Microsoft SQL SERVER2000。中间层Servlet为JudgeServlet，系统的流程包含六个步骤：

（1）客户端提出读取数据库数据申请，数据库数据申请提交给中

间层的JudgeServlet；

（2）对客户端的申请进行数字签名验证，确认其合法性；

（3）中间层JudgeServlet构建一个SQL语句，将SQL语句提交给JDBC；

（4）远程数据库服务器执行SQL语句并将查询结果返回给中间层JudgeServlet；

（5）中间层JudgeServlet得到查询结果信息，对结果利用MD5算法计算消息摘要,把结果和消息摘要形成一个对象，序列化后返回给客户端的Applet；

（6）客户端的Applet对返回结果进行处理，分离出消息摘要和查询结果信息，进行消息验证，验证通过后则进行图形处理，否则要求重新发送数据。

目前，在企业内部网和宽带网络环境下，基于上述流程的远程股票信息系统的速度还是比较快的，由于我们在系统实现过程中采用了数字签名和消息摘要技术，因此可以保证数据信息的安全性和合法性，从而最终使得系统能够安全稳定的性能。

参考文献

1 李洪宝、曾文方等，基于web的实时信息发布系统的设计与实现[J]，计算机应用，1999.12 27~28。

2 (美)Bruce Eckel著，候捷译，Java 编程思想(第2版)[M]，机械工业出版社，2002。

3 http://java.sun.com/j2ee/

4 蔡剑、景楠，Java 网络程序设计，J2EE（含1.4最新功能）[M]，清华大学出版社，2003。

5 (美)John Bell Tony Loton. Java Servlets 2.3编程指南[M]，电子工业出版社，2002。

6 (美)Jess Garms著，庞南等译，Java安全性编程指南[M].北京:电子工业出版社，2002。

7 (美)Joseph J.Bambara等著,刘锐业纫鼍引2EE技术内幕[M]，机械工业出版社，2002。

8 (美)Li Gong 著，JAVA 2 平台安全技术——结构、API 设计和实现[M]，机械工业出版社，2000。

9 (英)Danny Ayers等著,曾国平等译，Java服务器高级编程[M]，机械工业出版社，2001。

10 左黎明、盛梅波、艾剑峰，JAVA技术在远程实时数据的图像处理中的应用,[J]:计算机系统应用，2003.8。