一个新的电子商务的安全方案:HTML表单签名*)

A New Resolution of Safe E-Commerce:HTML Form Signing

李 娟 左 春

(中国科学院软件研究所 北京100080)

Abstract Many kinds of e-commerce require the ability to prove that someone has authorized a transaction. One way to provide such proof is to associate a digital signature with data generated as the result of a transaction. However, until now, there has been no widely available, standards-based way to digitally sign forms or other transaction-related text on the Web. This paper presents an implementation of HTML form signing, which utilizes the Public-Key Cryptosystem and the Digital Signature technology. By providing components of client and server, the system signs the user's form at the client end and verifies it at the server end.

Keywords HTML Form, Form singing, ActiveX Control

随着计算机和电子通讯技术,包括因特网的迅猛发展,金融电子化的步伐大大加快,许多传统上基于纸面的、常常需要签名盖章的重要凭证,已陆续转化为数字电子媒体的形式出现。目前,许多网站都是通过HTML表单(或称为HTML页面)的形式在Internet上接受用户输入的请求信息,从非常简单的订阅电子杂志的请求到具有高度商业价值的信息,如BtoB、BtoC业务中用户和商家之间传递的商业信息。在交易过程中为双方提供不可伪造的、不可抵赖的签名信息将是非常必要的。虽然SSL技术在SSL连接期间提供暂时的客户身份认证,但它不能为连接期间发生的交易提供持续稳定的认证。一个提供这样认证的方法是把数字签名连同交易产生的数据作为交易的结果信息。

1 签名交易信息方法之分析比较

由于普通的HTTP协议不支持数据的加密与签名,信息以非加密的形式在网络上传播,这就可能被非法窃听、非法篡改,没有签名信息,又无法确认信息发送者的真实性,无法防止信息发送者抵赖交易。为了解决这个安全漏洞,目前,实现对传输信息进行加密和签名的方法主要有以下几种:

(1)采用HTTPS协议 SSL(安全套接层)是Netscape公司为通过互联网传送保密文档而开发的协议,SSL技术使用公开密钥的算法将要通过SSL连接传送的信息进行加密,保证传输的安全性。HTTPS加密协议就是通过SSL进行HTTP传输的协议,它是一种最通用的方式,目前Navigator和Internet Explorer都支持SSL。但是这种方法有以下的缺点:

*)本文得到国家863计划项目(863-306-ZD06-02-8)的资助。李 娟 研究生,研究方向:软件工程。左 春 研究员,研究方向:软件工程。

题。

参考文献

1 Moukas A. Robert Guttman and Pattie Macs, Agent-mediated Electronic Commerce: An MIT Media Laboratory Perspective. Software Agents Group, MIT Media Laboratory

2 Nwana H S, et al. Agent-Mediated Electronic Commerce: Issues, Challenges and some Viewpoints. In: Proceedings of Autonomous Agents'98

3 Tsvetovatyy M, et al. MAGMA: An Agent-Based Virtual Market for Electronic Commerce

4 Chavez A, Maes P. Kasbah: An Agent marketplace for buying and selling goods. 1996

5 Dasgupta P, et al. A Supplier-Driven Electronic Marketplace Using Moblile agents. Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA

6 Zacharia G, et al. An agent system for comparative shopping at the point of sale. Software agents Group, MIT Media Laboratory

7 Wurman P R, Wellman M P, Walsh W E. The Michigan Internet AuctionBot. A Configurable Auction Server for Human and Software agents. University of Michigan, Artificial Intelligence Laboratory

首先，SSL 技术的设计主要是用来保证数据的加密传输，虽然可以通过获得认证证书 CA 来保证客户连接的服务器没有被假冒，但服务器依然不能认证连接客户的身份，即使使用双向 SSL，也只能保证本次 SSL 会话中数据的完整性及其不可抵赖性，而无法实现持续稳固的客户端身份验证。

其次，SSL 安全套接层位于传输层和应用层之间，是一种基于链路的加密方式，对应用程序是透明的，因此，处理交易请求的 Web 处理程序依然无法得到客户端的签名信息，很难确认客户的身份。

(2)采用安全代理/安全网关的方式　这种方式通过分别提供客户端安全代理和服务端安全网关，在它们之间采用 HTTPS 协议进行通讯，保证了数据在代理和网关之间传输的安全性。当用户发出的 HTTP 请求到达客户端代理时，客户端代理会判断用户发出的请求包中的 URL 资源是否为受到保护的资源而决定是否建立安全连接或是否对数据进行签名操作。这种方式依然存在很多缺陷：

首先，在这种方式中，客户端安全代理需要判断哪些信息需要签名，因此必须分析浏览器传过来的数据流，当页面需要更改时，将危及到客户端安全代理的分析，有时甚至需要更新客户端安全代理程序。

其次，由于对传输信息的验证和客户端身份的认证依然是在服务端安全网关，因此 Web 服务器上的应用处理程序还是无法直接获得客户的签名信息和身份信息。

最后，由于这种方式需要在浏览器和 Web 服务器间增加一个应用层，极大地增加了系统的复杂度，从而带来系统的不稳定性，影响了系统的性能，加上安全代理/安全网关需要安装和配置，也就增加了系统维护的工作量。

针对以上现状，我们提出了下述的一种方法，能有效地解决上述问题。

(3)页面中嵌入 ActiveX 控件的表单签名方法　以上两种方式都是在安全套接层完成对传输信息的签名操作，Web 服务器端的应用处理程序不能很好地和它们结合起来。为了解决上述方式的缺点，我们想到采用以 MICROSOFT 的 DCOM 技术为核心的 ActiveX 技术，在 ActiveX 中结合数字签名技术，实现端到端的安全解决方案。它具有以下优点：

首先，由于它采用端到端的思想，信息在传输之前就进行签名，传输过程不作任何改动，因此对签名信息的验证也放到了 Web 服务器端的应用处理程序，应用处理程序基本不用做大的改动，只要先对签名信息及客户身份进行验证，然后继续常规的处理。这种方式与原有系统的结合非常紧密。

其次，ActiveX 控件的功能由 HTML 页面中的 Java Script 语句控制，在页面需要更新时只要相应更改 Script 语句即可，提高了系统的适应能力；这种方法也没有改变原有系统的结构和传输方式，因此对系统的性能基本没有影响。

最后，ActiveX 控件可以随 HTML 页面同时下载、自动安装，并且可以一次下载反复使用，从而减轻了系统维护的工作量。

2　表单签名系统的具体设计实现过程

在我们的表单签名系统中，利用了公开密钥算法和数字签名技术的思想。现分别叙述如下：

基于密钥的算法通常分为两大类：对称算法和公开密钥算法。在对称算法中，用于加密的密钥和用于解密的密钥是完全相同的，因此算法的安全性依赖于密钥，带来传送和保管密钥的问题。在公开密钥算法中(如 RSA 算法)，加密密钥不同于解密密钥，加密密钥公之于众，称为公钥，解密密钥只有解密人自己知道，称为私钥。相对于对称算法而言，公开密钥算法除了能保证信息的保密性、完整性和安全性外，它还能对信息发送人的身份进行验证。

数字证书是在用户公钥后附加了用户信息及 CA 的签名。公钥加密的信息只能由与之相对应的私钥解密。为确保只有收件人才能阅读自己的信件，发送者要用收件人的公钥加密信件，收件人便可用自己的私钥解密信件。同样，为证实发件人的身份，发送者要用自己的私钥对信件进行签名，收件人可使用发送者的公钥对签名进行验证，以确认发送者的身份。有了数字证书，网上安全才得以实现，电子邮件、在线交易和信用卡购物的安全才能得到保证。

数字签名就是信息发送者用其私钥对报文中提取出的特征数据(或称数字指纹)进行私钥的加密，以保证发信人的真实身份，并可防止其抵赖曾发过该信息，同时也确保信息报文在传递过程中未被篡改，当信息接收者收到经签名的报文后，就可以用发送者的数字证书对数字签名进行验证。

我们的 HTML 表单签名系统利用以上两种思想，从总体结构上分为两个部分，一部分是与客户端浏览器配合使用的组件，另一部分是与服务器端(Web 站点)表单处理程序相配合使用的 API。

客户端我们采用 ActiveX 控件(称为 Signing Component)来实现对用户在表单上输入的信息的签名。它包含一组对表单数据进行签名操作的函数，在使用时，这些函数将会被一段 HTML 页面中的 JavaScript 代码调用，实现对表单中数据的签名操作。Signing Component 可以通过浏览器自动下载到客户

使用的计算机系统中，也可以使用一个单独的安装程序进行安装。

服务器端提供一组函数（称为 Verification API），可以在已有的表单处理程序中调用这些函数，实现对签名信息的验证及对签名用户的确认。由于目前 Web 服务器程序开发环境较多，因此我们提供了多个不同的版本，如 Java 版、CGI 版、ASP 版等。

HTML 表单签名和验证的大致过程（图1）是这样的。

首先我们在需要签名的 HTML 页面中插入一段 JavaScript 代码，用于确定表单中需要签名的字段。当用户的浏览器接收到该 HTML 页面后，在用户填写完数据并提交表单时，浏览器自动执行 JavaScript 代码得到需要签名的信息，JavaScript 代码调用 Signing Component 中的签名功能，将需要签名的信息传给相应的签名函数，该函数从用户 IC 卡或其他设备中读取用户私钥，并用该私钥签名信息，然后返回 PKCS #7 格式的签名数据包，JavaScript 代码再将返回的数据包放到 HTML 页面中的一个 HIDDEN 域里，提交给服务器。当服务器端的表单处理程序接收到用户的请求信息时，它将首先调用 Verification API 验证客户端的签名信息，如果验证通过，接着处理请求信息，否则用户的请求信息被视为无效。

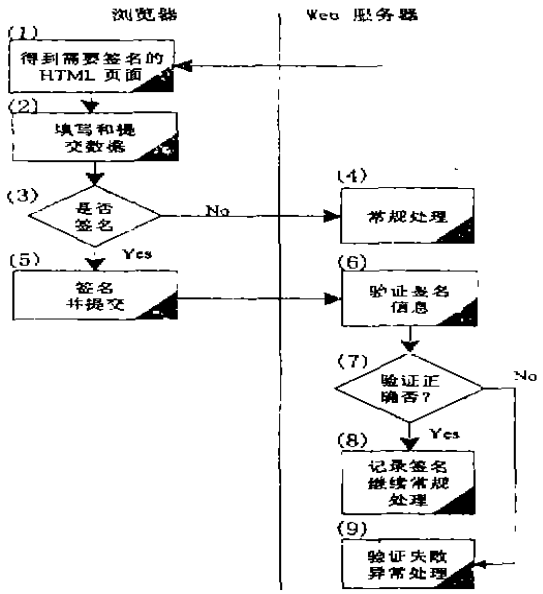

图1　表单签名和验证的过程示意图

2.1　客户端签名技术的实现

目前最通用的浏览器 Internet Explore 4.0或以上版本都支持 Java Script 和在网页中嵌入 ActiveX 技术，因此我们在表单签名系统的客户端实现中分别应用以上技术，下面我们分两部分来介绍。

（1）签名功能的实现　签名功能通过 ActiveX 控件（Signing Control）来实现。该控件将被插入到 HTML 页面中，并由该页面中的 JavaScript 脚本控制。该控件实现签名功能的主要接口是：

BSTR SignText（LPCTSTR pMessage，long Length）

参数 pMessage 是指向需要签名的数据的指针，参数 Length 是需要签名的数据的长度，函数的功能为：使用用户的私钥对数据进行签名，并返回签名后的数据包。由于签名后的数据要在 Internet 上传输，因此我们对签名后的数据进行了 Base64编码。

为了实现上述的签名过程，控件需实现以下功能：

1. 从用户的 IC 卡或其他介质中取得用户的私钥。如果是带 CUP 的 IC 卡，可以由 IC 卡本身完成签名算法，否则就从 IC 卡或其他介质读取用户的私钥。

2. 验证用户证书。验证用户证书，包括验证证书是否是由服务器端认可的 CA 签发的，以及证书的有效期。

3. 对签名数据进行 PKCS #7打包。我们采用 RSA 实验室推荐的加密信息语法标准 PKCS #7保存签名信息，可最大程度保证通用性。

4. 编码 PKCS #7签名数据。首先需要对 PKCS #7签名数据进行 ANS.1编码，再对编码后的数据进行 Base64编码。

5. 用户界面的实现。包括提供用户选择私钥的选择框和取得用户私钥保护口令的输入框等。

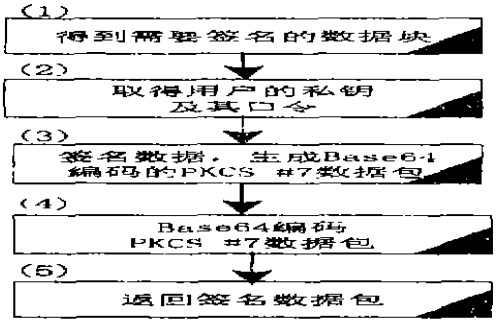

图2　签名过程示意图

（2）HTML 页面和 JavaScript 代码　首先我们需要将 ActiveX 控件插入到 Form 页面中，可以通过插入以下代码来实现：

```
<object id="SignControl"
    classid = "CLSID: CE7C3CF0-4B15-11D1-ABED-
              7u9549C10000"
    codebase="http://SignControl.cab"
    type="application/x-oleobject">
</object>
```

在页面中插入该段代码后，后面的 JavaScript 代

码中就可以用 SignControl 来引用该控件的功能，codebase 后的代码是当用户本机没有安装该控件时，浏览器会自动从该地址下载并安装该控件。JavaScript 代码如下：

```
function signForm(theForm, theWindow, validation) {
    var formSize = theForm.elements.length;
    var elem,
    var signedText,
    for(var i=0, i<formSize; i++) {
        elem = theForm.elements[i];
            text+=(elem.name + "=" + elem.value);
    }
    text+="</UL>"
    SignControl.SignText(text, signedText);
    validation.value = signedText;
    SignControl.SignText
    return true;
}
```

我们将 signForm 函数放在 Form 的 ONSUBMIT 属性中，每当用户填完数据按下 Submit 按钮时，该段代码将自动执行，将用户填写的数据合成一个字符串，并签名该字符串，将签名后的数据赋给 HIDDEN 域 validation，随后浏览器将把所有数据提交到 Web 服务器上。

2.2 服务器端验证技术的实现

在服务器端我们提供一组签名数据的验证函数（Verification API），服务器端的请求处理程序通过调用这组 API 函数来验证 PKCS#7签名数据包，并确认签名者的个人信息。Verification API 中对外接口主要包含三个函数，分别是：

（1）BOOL SetCACert(LPCTSTR pCACertFile)：参数 pCACertFile 是指向 CA 根证书文件名的指针，CA 根证书是 DER 编码的 X509证书。该函数的功能是读取 CA 根证书，并保存到内存中，用于验证客户端上传的签名信息中的证书的有效性；

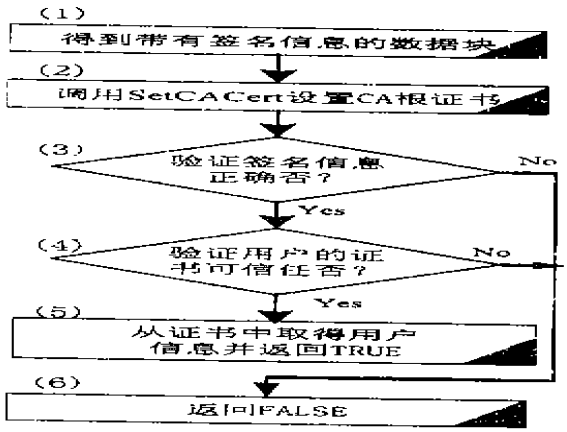

图3 验证函数处理过程示意图

（2）BSTR VerifyData(LPCTSTR pMessage, long Length)，参数 pMessage 是指向需要验证的签名数据的指针，参数 Length 是签名数据的长度，该函数的功能为先对签名数据进行 Base64解码，然后验证数据是否被篡改，如数据验证通过，再用刚才读取的 CA 根证书验证签名人的证书是否是可信任的证书，如验证通过，则返回用户签名前的数据；

（3）BSTR GetSignatoryInfo()：该函数的功能是返回签名人的信息，如姓名、单位、eMail 地址等，这些信息是保持在签名人的证书中的，因此，如果签名数据验证通过即可确认签名人的信息。

3 实际应用及结论

上面讨论的表单签名技术可以实现用户的签名及服务器端对用户及签名信息的验证，通过几个大型单位的试用，具有很好的性能和实用价值，在实际应用中，还需要考虑以下问题：

1. 签名数据在网上的安全传输，目前我们可以使用 SSL 技术来保证签名数据的加密传输；

2. 签名数据在服务器端和客户端的保存。为了在日后还能查询本次交易的细节和实现防抵赖，我们可以在服务器端将签名数据保存到数据库中，客户端也可以将签名数据保存到文件或数据库中；

3. 用户签名用的数字证书的申请。一般商务网站不可能建立自己的 CA，因此可以从一些已经得到公认的 CA（如 CFCA）申请；

4. 客户端私钥和证书的保存。在安全性要求很高的应用中，建议将私钥存放到带 CUP 的 IC 卡或 iKey 中，但在某些情况下也可以将私钥保存在网上，用户可以随时在线获取自己的私钥，这样能给用户带来极大的方便性。

参 考 文 献

1 Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. Wiley & Sons, Inc. 1996

2 Man Young Rhee. Cryptography and Secure Communications. Mc Graw-Hill Book Co. 1994

3 Linn J. Privacy Enhancement for Internet Electronic Mail: Message Encryption and Authentication Procedures. RFC 1421, DEC, 1993-02

4 Netscape Communications Corporation. URL: http://developer.netscape.com/tech/security/formsign/formsign.html