

XML 数字签名在工作流系统中的应用

傅德胜, 王 强

(南京信息工程大学 计算机与软件学院, 南京 210044)

(wongqiang@yeah.net)

摘 要:针对工作流系统中存在的多重签名以及对文档进行较细粒度的签名需求,提出了“签名之上的签名”的机制,建立了以该机制为核心的 XML 数字签名在工作流系统中的应用模型。该模型通过对待签名的文档转化为 XML 数据,方便了系统对待签文档的处理。在对 XML 文档的处理进程中,各处理节点在前任处理节点的基础上对待签 XML 文档进行验证和签名。最后开发了采购审批工作流系统,并通过一个典型的采购审批场景验证了该模型的正确性和有效性,为 XML 数字签名在工作流系统中的应用提供了可行的解决途径。

关键词:XML 数字签名;工作流;多重签名;部分签名;XPath

中图分类号:TP309.2 **文献标志码:**A

XML digital signature application in workflow system

FU De-sheng, WANG Qiang

(School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing Jiangsu 210044, China)

Abstract: In view of the demand for multi-signature and fine-grained signature in workflow systems, the authors proposed the "Signature on Signature" mechanism and put forward an application model of eXtensible Markup Language (XML) signature in workflow systems. In this model, the document to be signed was converted to XML format, and this facilitated the handling of the document for the system. In the processing of the XML document, every processing node signed and verified on the basis of the former processing node. In the end, a purchase approval workflow system was developed. Taking a typical purchase approval scenario for example, the validity and effectiveness of the presented model were verified, and a feasible solution was provided for the application of XML digital signature in workflow systems.

Key words: eXtensible Markup Language (XML) digital signature; workflow; multi-signature; partial signature; XPath

0 引言

在传统的纸质工作流中,签名是必不可少的环节。通常,一份文件的起草到最后的生效往往会涉及多个部门的审核签字或者盖章。随着信息技术的发展,越来越多的业务基于网络平台开展,形成了工作流管理系统。在工作流管理系统中对涉及到的数字文档进行签名依然是保障文档的完整性和不可否认性的一个重要环节。

工作流系统对文档进行数字签名有一些特别之处,比如一个事务处理过程中需要多个用户对同一个文档进行签名,即所谓的多重签名^{[1]302},这和通常所熟悉的单个签名不同的;另外,在有些用户对文档进行签名时,可能只对文档的一部分感兴趣,也就是实现部分签名。对于多重签名,也许可以通过改进现有的签名算法加以实现,但是,以传统的数字签名技术实现部分签名往往比较困难。

XML(eXtensible Markup Language)技术近年得到迅速发展,利用 XML 技术可以实现数据在异构网络下企业间不同系统间共享和交换。基于 XML 的数字签名技术早在 2001 年就成为了 W3C 标准^{[2]2,[3]}。XML 文档作为高度结构化的数据,具备很好的扩展性。在应用中,可以将待签名的文档转换为 XML 数据,然后对该 XML 文档签名。本文主要讨论将基于 XML 的数字签名技术应用在工作流系统中时要解决的问题。

1 基于 XML 的数字签名技术

1.1 传统的数字签名技术

传统的数字签名技术基于目前广泛使用的公共密钥体

系^{[4]21-25}(如图 1、2 所示)。发送方首先用某种消息散列算法(如 SHA-1、MD5 等)对要签名的文档进行散列运算,得到一个固定长度的消息摘要。然后用自己的私存密钥对摘要进行加密形成发送方的数字签名,并将这个数字签名作为报文的附件和报文一起发给接收方。接收方首先从接收到的原始报文中用同样的算法计算出新的消息摘要,再用发送方的公钥对报文附件的数字签名进行解密,比较两个消息摘要,从而验证签名确实来自发送方。鉴于私钥的私密性,任何通信第三方企图用自己的私钥对发送报文进行加密或者在报文发送过程中对报文做了修改,都将导致签名验证的失败^{[4]28-31}。

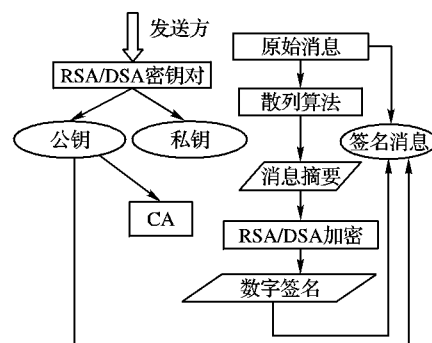


图1 数字签名的生成

传统数字签名能满足数据完整性、不可否认性、身份识别和认证等基本的安全要求^{[4]6}。然而传统的数字签名技术的不足在于签名粒度过粗,签名对象是文档整体。

收稿日期:2010-09-10;修回日期:2010-10-29。

作者简介:傅德胜(1950-),男,江苏靖江人,教授,博士生导师,主要研究方向:信息安全;王强(1985-),男,江苏泰州人,硕士研究生,主要研究方向:信息安全。

1.2 基于 XML 的数字签名

基于 XML 的数字签名技术本质上和传统的数字签名技术一样。但是,具备一些传统数字签名不具备的优点。其语法定义结构如下,结构中各标记的含义可参见文献[2]。

```
<Signature ID? >
  <SignedInfo >
    <CanonicalizationMethod />
    <SignatureMethod />
    ( <Reference URI? >
      ( <Transforms /> )?
      <DigestMethod />
      <DigestValue />
    </Reference > ) +
  </SignedInfo >
  <SignatureValue />
  ( <KeyInfo /> )?
  ( <Object ID? /> ) *
</Signature >
```

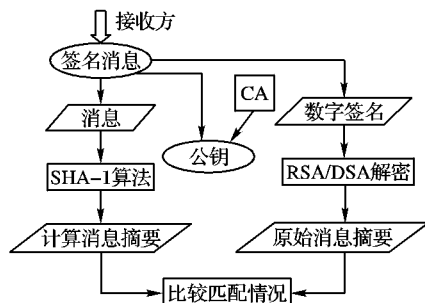


图2 数字签名的验证

XML 数字签名充分利用了 XML 本身强大的表达能力和扩展能力,不仅可以像传统的数字签名技术一样对整个文档签名,还可以实现在较细的粒度上对文档的特定部分进行签名。在 XML 签名的 <Reference> 元素中,其 URI 属性不仅可指定本地或网络上的文本或二进制数据,还可指定 XML 文档内部的某个元素。如果想对多份数据签名,可以采用多个 Reference 元素分别指向不同的签名对象^[5]。

一个 XML 签名核心验证过程必须完成:1) 引用验证,指对包含在 <SignedInfo> 元素内每个 <Reference> 中摘要的验证;2) 签名验证,指对包含在 <SignatureValue> 元素内加密签名的验证。具体的引用验证和签名验证步骤参见文献[2]。

1.3 多重数字签名的可行性

在 workflow 系统中应用数字签名通常会涉及到多人对同一份文档进行数字签名。通常有两种方法来处理多重签名,第一种是使用复杂的数学模型,比如改进的 RSA 模型^{[6]6037}。这些模型本身结构上非常复杂,扩展性差,很难应用到现有的一些 workflow 系统中。第二种方法是以一种预先定义的对一份文档多次应用单个数字签名,这种方式的优点是不需要用复杂的数学公式,但是需要在签名和验证的过程中分割出文档的各个部分,否则签名的验证过程会出错。

XML 是高度结构化的数据,如果在 workflow 中将待签名的数据转换成 XML 文档,在处理过程中就很容易借助 XML 相关技术分离出 XML 文档的相关部分进行签名和验证。然而,仔细观察 XML 数字签名的语法结构似乎并不支持多重数字签名。多重签名是由多人组成的群体签署同一份文件,而上述 Signature 定义中,一个 SignatureValue 可以对应多个文件 Reference,却没有一个 Reference 对应多个 SignatureValue 的结构以适应多重签名的需要^{[2]7}。虽然如此,经过分析不难发现,基于 XML 的数字签名语法不直接支持多重签名,但是 XML 数字签名规范并没有限制 XML 文档中包含的 Signature

元素的个数,因此,完全可以在待签的 XML 文档中包含多个 Signature 元素代表对同一份文件的多个人员的签名。文献[1]也提到了基于这种方式的多重签名的可行性。

2 XML 数字签名在 workflow 系统中的应用模型

基于 workflow 系统的需求,本文设计了一种多重签名方案,即按预先确定的签名顺序多次使用单一数字签名,由此成为一个组合的多重签名方案。

2.1 有序多重签名

以一个物资采购审批流程为例,在通常的工作流程中,各部门对采购计划的审批顺序往往是固定的。在传统的纸质工作流程中,也许无法保证签名的顺序性,但是在 workflow 管理系统中,就能保证部门审批是按序执行的^{[6]6036}。当 workflow 的某一个环节对 XML 文档进行签名后,一个该处理节点的签名便会附在原文件上,成为审批单的一部分。当 workflow 进行到下一个环节时,该环节的用户首先验证上一个环节签名的有效性,然后再依次验证前面所有各个环节签名的有效性。

考虑这样一个审批场景,三个用户: Amy、Bob、Clare 按序对一份文档 M 审批签名。

1) Amy 用自己的私钥对文档 M 进行签名,本文把签名后的文档记为 SA。

2) Bob 收到 M 和 SA 后,首先用 Amy 的公钥验证 Amy 签名的有效性,然后他再用自己的私钥在 Amy 签名的基础上再对 XML 文档进行签名。这时候文档中存在两个签名, Amy 的签名和 Bob 的签名,把该文档记为 SB。

3) Clare 收到 M 和 SB 后,首先用 Bob 的公钥验证 Bob 签名的有效性,如果必要,他还可以再次用 Amy 的公钥再次验证 Amy 签名的有效性。

这种有序多重签名方案拓扑如图 3 所示,由于各个处理节点是在前面各个处理节点的基础上再次签名,本文把这种有序多重签名机制称为“签名之上的签名”(Signature on Signature)。

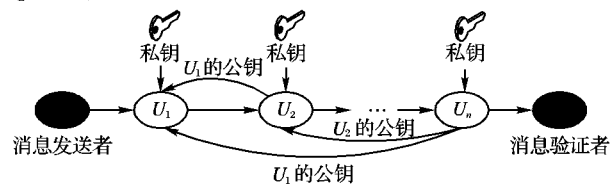


图3 有序多重数字签名

2.2 应用模型

以一个典型的审批 workflow 系统为例,流程执行分为三个阶段,即流程的启动、流程的处理、流程的结束^[7]。首先由流程的发起人发起审批流程,通常情况下,该用户起草一个审批单,填写审批内容后,系统负责将该 Web 表单的内容转换为 XML 文档,并且用该发起人的私钥进行签名,最后将 XML 文档持久化到数据库中。流程启动后,模型进入流程处理阶段。流程会向下一节点的处理人员发送流程处理任务。

下一节点处理人员登录系统后,通过接收到的任务处理流程。首先该节点的处理人员从数据库取出 XML 文档,并将该文档还原为 Web 页面,以可视化的方式显示审批单的详情^[8],以及各个节点对该审批单的签名情况。然后,该处理人员需要依次验证前面若干个节点人员签名的有效性。验证签名的顺序按签名顺序的逆序进行,即首先验证第 n 个处理节点的签名,然后是第 $n-1$ 个处理节点直到第 1 个节点。本文在对 XML 做数字签名的过程中,会直接在数字签名的 KeyInfo 节点中包含公钥信息,这样在对数字签名做验证的时候便可直接用该公钥验证签名的有效性。然而,本文不应该

依赖于该公钥信息,还应该利用签名中的用户 ID 信息,到 CA 服务器上取得用户的公钥信息,然后和 KeyInfo 的公钥信息进行对比,如果不一致,即使签名验证通过,该签名也是无效的^[9]。一旦 XML 签名验证失败,应该立即抛出错误,终止工作流的执行。XML 数字签名在工作流系统中的整体应用模型如图 4 所示。

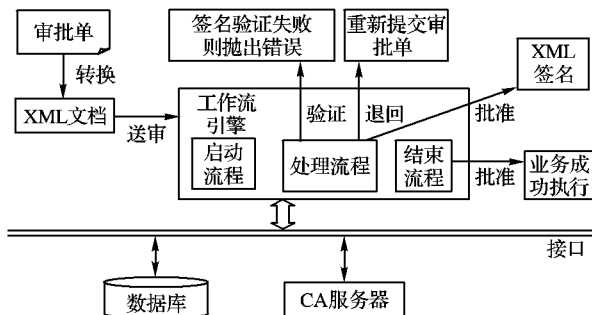


图4 XML签名在工作流中的应用模型

XML 签名验证通过后,节点处理人员开始对审批单进行审批,在填写处理意见后,做出相应处理(如通过或者退回)。在最后一个节点用户处理完毕后,如果审批通过,则会进入流程的第三个阶段——流程结束阶段。至此对 XML 文档的处理结束,生成最终版本。发送消息给流程发起人,整个工作流程完结。

3 方案的实现

3.1 开发平台

鉴于 XML 数字签名的种种优点,各大软件厂商几乎都对 XML 数字签名提供了很好的支持^{[1]301}。例如,IBM 的 AlphaWorks 小组开发了 XML 安全套件(XMLSecuritySuite),提供 XML 数字签名的函数库,微软 System. Security. Cryptography. Xml 命名空间里的 Signature 类是完全参照 W3C 和 IETF 所提出的 XML 数字签名标准设计的。首先基于 ASP.NET 开发了 B/S 架构的采购审批系统,其核心包含角色与授权控制、单据制作与工作流引擎(基于微软的 WWF (Windows Workflow Foundation) 工作流引擎^[10]),融入了数字证书及签名技术。另外,该采购审批系统可以灵活地实现审批流程的自定义。该系统采用三层架构:表现层为 WebForm 窗体,负责与用户的交互;业务逻辑层负责节点事件处理、XML 文档的可视化、文档签名等业务;数据层负责提取数据,返回记录集。

3.2 应用场景

以一个自定义的审批场景为例,如图 5 所示。首先系统管理员发起审批流程,然后依次由科长、处长、财务科长进行审批,流程中各种状态转换如图 5 中的箭头所示。整个工作流起始于系统管理员提交采购审批单。流程中涉及到的审批单以 XML 形式展现,当工作流进行到某一个环节时,首先由该环节的处理人员验证之前各环节处理人员所做的签名的有效性,一旦发现签名无效,抛出错误,终止工作流。验证通过后,再由该处理人员对 XML 形式审批单进行签名,并向下一个流程处理人员发送处理任务。

3.3 签名的生成过程

鉴于在工作流中各个环节生成签名和验证签名的过程类似,这里以一个节点处理过程为例进行阐述。

3.3.1 数据库到 XML 文档的转换

首先从数据库中提取采购审批单数据信息,并将该数据信息转换成 XML 文档结构的形式。要将数据库中的数据转换为 XML 文档,必须首先生成 XmlDocument 对象,然后从数

据库中提取待转数据表的记录,并将记录数据通过字符串拼接的形式组合成 XML 文档的内容,最后调用 XmlDocument 对象的 LoadXml 方法生成 XML 文档。在本系统中待签名的 XML 文档如下所示:

```
<?xml version="1.0" encoding="gb2312"?>
-<ApprovalSheet xmlns="http://www.example.com">
  <ApprovalID>b67ed5bd-bc65-4cec-be7d-0602457c0ff6</ApprovalID>
  <Applicant>tom</Applicant>
  -<Items>
    -<Item>
      <ItemName>PC2 台</ItemName>
      <ItemValue>12 000</ItemValue>
    </Item>
    -<Item>
      <ItemName>打印机 1 台</ItemName>
      <ItemValue>12 000</ItemValue>
    </Item>
  </Items>
  <Total>13200</Total>
  <DateOfApplication>2010-09-30 11:22:51</DateOfApplication>
  <Remarks>办公耗材</Remarks>
</ApprovalSheet>
```

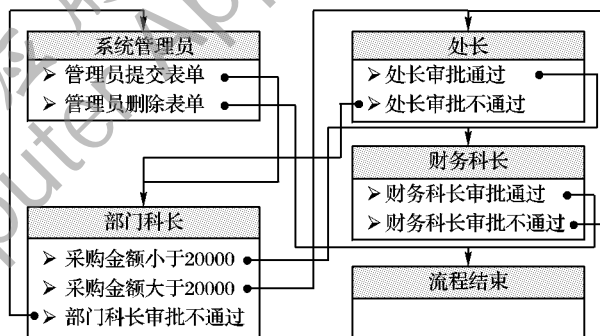


图5 采购审批工作流状态机

3.3.2 验证先前数字签名的有效性

在对 XML 文档应用自己的签名之前需要验证先前处理节点的数字签名的有效性。如果发现某一个节点的数字签名无法通过验证应该立即终止工作流,并抛出错误。用以验证 XML 数字签名有效性的方法原型是: bool VerifyMessage(XmlDocument SignedXml, string UserID),其中 SignedXml 是待验证的 XML 文档,UserID 是待验证的签名者的 ID。该方法返回一个 bool 值,指示签名验证是否成功。该方法的内部实现如下。

1) 首先创建一个 XmlDocument 对象,将已签名的 XML 文档以 Load 方法装入。

2) 创建一个 SignedXml 对象,并将待验证的签名装入:

```
XmlElement sigElt = (XmlElement) messageToVerify. SelectSingleNode(
  ("/* *[@Id='Signature' + SignerID + '']", nsm);
SignedXml sig = new SignedXml(messageToVerify);
sig.LoadXml(sigElt);
```

3) 最后使用 SignedXml 对象的 CheckSignature 方法验证 XML 数字签名。

3.3.3 对 XML 文档应用签名

如果先前各节点签名验证无误,那么该节点用自己的私钥在之前签名文档的基础上再次进行签名,用以对 XML 文档进行数字签名的方法原型是: void SignMessage(XmlDocument DocToSign, RSA m_signingKey, string SignerName, string SignerID)。该方法的内部实现如下。

1) 创建一个 SignedXml 对象, 获取 RSA 密钥, 并将用于对 SignedXml 对象进行签名的不对称算法密钥设置为该密钥。

```
SignedXml sig = new SignedXml(doc);
RSA m_signingKey = (RSA) new RSACryptoServiceProvider();
sig.SigningKey = signingKey;
```

2) 创建一个 DataObject 对象, 用以封装签名者相关信息。

```
DataObject signer = new System.Security.Cryptography.Xml.DataObject();
sig.AddObject(signer);
```

3) 为签名对象添加引用, 并且为该引用添加适当的变换。本研究添加了两个引用, 第一个引用代表对 XML 全文进行签名 (不包括签名部分的数据)。第二个引用本文只对 DataObject (包含在签名中) 中的 SignerID 签名, 为了从整个 XML 文档中选取某一个节点, 本文应用了一个 Xpath 变换^[11], 将 SignerID 元素从 XML 文档中选取出来进行签名。具体如下:

```
Reference docRef = new Reference("");
docRef.AddTransform(new XmlDsigEnvelopedSignatureTransform());
Reference signerRef = new Reference("#Signer" + signerID);
string_m_xpathTransString = "<XPath xmlns:my='http://example'>ancestor-or-self::my:SignerID</XPath>";
XmlDsigXPathTransform xpathTrans = new XmlDsigXPathTransform();
xpathTrans.LoadInnerXml(doc.ChildNodes);
signerRef.AddTransform(xpathTrans);
```

4) 创建一个 KeyInfo 对象, 添加密钥信息:

```
KeyInfo ki = new KeyInfo();
ki.AddClause(new RSAKeyValue(signingKey));
sig.KeyInfo = ki;
```

5) 最后使用 ComputeSignature 方法生成 XML 签名, 并以 XML 文档格式保存, 完成 XML 数字签名过程。

```
sig.ComputeSignature();
```

4 结语

本文主要探讨了 XML 数字签名在工作流系统中的应用, 解决了工作流系统中存在的多重签名以及对文档进行较细粒

度的签名需求, 提出了“签名之上的签名”的机制, 给出了 XML 数字签名在工作流系统中的应用模型。基于给出的模型, 将 XML 数字签名应用在一个采购审批工作流系统中, 取得了较好的效果。在未来的研究中将进一步完善该模型, 消除模型中可能存在的一些安全隐患。

参考文献:

- [1] 张明哲, 范俊逸. XML 数字签名标准与应用[J]. 电信研究, 2003, 32(2): 299-306.
- [2] XML-signature syntax and processing [EB/OL]. [2010-07-08]. <http://www.w3.org/TR/2001/PR-xmlsig-core>.
- [3] Canonical XML W3C recommendation [EB/OL]. [2010-07-19]. <http://www.w3.org/TR/2001/REC-xml-c14n>.
- [4] 赵泽茂. 数字签名理论[M]. 北京: 科学出版社, 2007.
- [5] 郭亮乐, 赵正德, 于清华, 等. XML 数字签名技术的研究与实现[J]. 计算机工程与设计, 2005, 26(5): 1211-1213.
- [6] LEUNG K R, HUI L C. Multiple signature handling in workflow systems [C]// Proceedings of the 33rd Hawaii International Conference on System Sciences. Washington, DC: IEEE Computer Society, 2000: 6033-6041.
- [7] 徐全生, 卢丙峰. 带有图片签名的工作流技术[J]. 沈阳工业大学学报, 2009, 31(4): 466-470.
- [8] KUBBILUN W, CAJEK S, PSARROS M, et al. Trustworthy verification and visualisation of multiple XML-signatures [C]// Proceedings of the 9th IFIP TC-6 TC-11 International Conference on Communications and Multimedia Security, LNCS 3677. Berlin: Springer, 2005: 311-320.
- [9] 蔡爽, 杜平安. ERP 工作流引擎中的数字签名技术[J]. 计算机应用研究, 2007, 24(4): 130-132.
- [10] KITTA T. Windows workflow foundation 高级编程[M]. 陈宇寒, 译. 北京: 清华大学出版社, 2008.
- [11] XML Path Language (XPath) [EB/OL]. (1999-11-16)[2010-06-29]. <http://www.w3.org/TR/Xpath>.
- [12] 王志晓, 吕林涛, 闫文耀. 基于 ASP.NET 技术和工作流模型的网上审批系统[J]. 计算机工程, 2004, 30(17): 83-85.

(上接第 807 页)

4 结语

本文针对 RFID 系统安全性问题, 提出了一种基于 Hash 函数和密钥阵列的安全认证协议。新协议使用标签 ID 的 Hash 函数值进一步保护标签的隐私, 每一对阅读器和标签之间都拥有独立的认证密钥可进一步防止假冒攻击, 可抵抗包括窃听、位置跟踪、重传攻击、拒绝服务和篡改等多种攻击方法, 对抵御来自系统内部的威胁具有明显的优势。新协议的复杂程度随系统中标签和阅读器数量的增加而加大, 且主要体现在服务器一端, 因为在实际系统中, 标签数量远远多于阅读器数量。而在计算机技术高度发达的今天, 服务器技术也不应该是实现该协议的瓶颈。

随着密码学技术的不断发展, 长时间使用相同密钥进行通信对该认证协议存在安全隐患, 但定时更新密钥会增加协议的复杂度, 更重要的是更新密钥需确保后台服务器、阅读器和标签的密钥同步更新, 一旦同步被破坏, 将会给整个系统带来灾难性的影响。这将是今后研究的重点内容。

参考文献:

- [1] SARMA S E, WEIS S A. Radio frequency identification: Secure risks and challenges [J]. RSA Laboratories Cryptobytes, 2003, 6

(1): 2-9.

- [2] OHKUBO M, SUZUKI K. Hash-chain based forward secure privacy protection scheme for low-cost RFID [C]// SCIS 2004: Proceedings of the 2004 Symposium on Cryptography and Information Security. Sendai: [s. n.], 2004: 719-724.
- [3] 余恬恬, 冯全源. 基于 Hash 函数的 RFID 挑战-应答认证协议[J]. 计算机工程, 2009, 35(24): 156-161.
- [4] 陈剑, 张春, 陈虹. 低功耗 RFID 的公钥密码系统实现[J]. 半导体技术, 2009, 34(9): 890-894.
- [5] PAISE R I, VAUDENAY S. Mutual authentication in RFID: Security and privacy [C]// Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2008: 292-299.
- [6] 张恒山, 管会生, 韩海强. RFID 系统中基于公钥加密的相互认证协议[J]. 计算机工程与应用, 2010, 46(5): 69-72.
- [7] 周永彬, 冯登国. RFID 安全协议的设计与分析[J]. 计算机学报, 2006, 29(4): 581-589.
- [8] 丁振华, 李锦涛, 冯波. 基于 Hash 函数的 RFID 安全认证协议研究[J]. 计算机研究与发展, 2009, 45(4): 583-592.
- [9] 白煜, 滕建辅, 张立毅, 等. 基于 Hash 锁的同步强化 RFID 验证协议[J]. 计算机工程, 2009, 35(21): 138-143.