信息和通信安全——CCICS'99

第一届中国信息和通信安全学术会议论文集

卿斯汉 冯登国 编

科学出版社

信息和通信安全——CCICS'99

第一届中国信息和通信安全学术会议论文集

卿斯汉　冯登国　编

科学出版社
2000

内 容 简 介

本书为第一届中国信息和通信安全学术会议论文集，收录论文 39 篇，内容涉及信息和通信安全的各个领域。主要内容包括：网络安全，访问控制，数据库安全，操作系统安全，电子商务，分组密码，公钥密码，数字签名和身份识别，密钥管理，安全协议，智能卡安全等。

本书可供从事信息安全、密码学、计算机、通信、数学等专业的科技人员和高等院校相关专业的师生参考。

图书在版编目(CIP)数据

信息和通信安全——CCICS'99：第一届中国信息和通信安全学术会议论文集/卿斯汉，冯登国编. -北京：科学出版社，2000

ISBN 7-03-007907-8

Ⅰ. 信… Ⅱ. ①卿… ②冯… Ⅲ. ①计算机网络-安全技术-学术会议-文集②计算机通信-安全技术-学术会议-文集 Ⅳ. TP393.53

中国版本图书馆 CIP 数据核字（1999）第 61920 号

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

中国科学院印刷厂 印刷

新华书店北京发行所发行 各地新华书店经售

*

2000 年 1 月第 一 版 开本：787×1092 1/16

2000 年 1 月第一次印刷 印张：16

印数：1—1 500 字数：365 000

定价：32.00 元

（如有印装质量问题，我社负责调换〈环伟〉）

第一届中国信息和通信安全学术会议
程序委员会

主　席：卿斯汉（中国科学院信息安全技术工程研究中心）

委　员：（以姓氏笔画为序）

丁存生　（新加坡国立大学）
王育民　（西安电子科技大学）
尹依群　（美国 RSA 实验室）
毛文波　（英国惠普实验室）
冯登国　（中国科学院软件所）
朱　洪　（复旦大学）
朱伟年　（香港 RACAL 公司）
李大兴　（山东大学）
李　宝　（中国科学院信息安全技术工程研究中心）
肖国镇　（西安电子科技大学）
何大可　（西南交通大学）
杨义先　（北京邮电大学）
何松涛　（中国科学院软件所）
杜　虹　（北京信息应用技术研究所）
来学嘉　（瑞士 R3 实验室）
吴文玲　（中国科学院信息安全技术工程研究中心）
南湘浩　（北京大学）
郭宝安　（清华大学）
龚奇敏　（信息产业部第三十研究所）
韩永非　（法国 GEMPLUS 公司）
裴定一　（中国科技大学研究生院）

前　言

展望21世纪，信息和通信安全面临更大的机遇和挑战。为反映我国学者在此领域的最新研究成果，为促进海内外华人在此领域的交流，国际信息和通信安全学术会议（ICICS）的地方版——第一届中国信息和通信安全学术会议（CCICS'99）于1999年12月8～10日在北京举行。本届会议共收到论文60余篇，每篇论文都通过三位专家评审和程序委员会讨论作出最终选择，本论文集收集了与信息和通信安全有关的实用理论和关键技术方面的论文39篇，内容涉及网络安全，访问控制，数据库安全，操作系统安全，电子商务，分组密码，公钥密码，数字签名和身份识别，密钥管理，安全协议，智能卡安全等方方面面。

我们衷心感谢所有投稿者对会议的关心和支持！衷心感谢程序委员会和论文评审专家的辛勤工作。

目　录

信息系统安全

认证协议和数字签名

信息对抗理论与技术

安全协议及其形式化分析

分组密码

公钥密码

逻辑函数

电子商务和智能卡

基于用户角色和阶段性控制的网上公文传送的安全机制

陈庆章
(浙江工业大学,杭州,310032)

摘　要　通过网络进行公文传送已经成为网络应用的重要组成部分。这种在网上进行的撰文、交办、批示等公文操作,因具有时效性、流动性和机密性,所以,除了要有效率地依照预定流程转移公文外,还必须具有严格的存取控制,以实现信息安全。

针对公文工作流程,本文提出一种基于用户角色和阶段性控制的公文传送安全机制,让与公文相关的群组可以在适当时间以指定的角色处理公文,不同的角色对公文拥有不同的处理权限,并依据用户角色、文件受保护的区域和公文传送进度,提出阶段性存取控制模型,实现公文安全传送。

关键词　网上公文传送　安全机制　存取控制　阶段性保护

一、网上公文传送的安全问题

公文生成和公文传送具有以下特性:

(1)公文文件由一群人分阶段制作完成,并且每一参与者只能在指定阶段处理文件。如承办人在撰稿阶段编辑基本数据;会签者在会签阶段签署意见;单位领导于批示阶段批示。

(2)每一参与者由于其角色(岗位)不同,对于公文文件的处理方式也有不同,角色划分如承办公文的撰稿者、拟办者、批示者等。

(3)不同角色只允许在文件中的部分数据区域进行编辑。如承办者于公文正文数据区;会签者于拟办签署数据区;决策者于批示数据区。

考虑到网络的特性,在网上传送公文时,如果对公文文件存取权不加以控制,就可能会造成下列问题:

(1)非撰稿者篡改公文各个数据区的数据。

(2)非拟办或批示者在签署区冒签。

(3)撰稿者在公文撰稿已确认并经他人签定意见或领导批示后,再修改基本数据区。

(4)拟办者在领导批示后,再修改拟办意见栏,让领导承担责任。

这些错误的行为可归纳为三类:角色性错误、范围性错误及阶段性错误。第一种情况是角色性错误,因为只有撰稿者才有权修改基本数据区;第二种情况是范围性错误,因为用户在自己没有改动权限的区域内操作;第三、四种情况是阶段性错误,一是撰稿者在拟稿或批示阶段并无权修改基本数据区,二是拟办者在批示阶段无权再修改拟办意见栏。因此,为了有效防止上述弊端,必须有一种安全的公文传送机制,否则,网上公文传送将难以实施。

关于安全机制的模型,已有众多的研究。见参考文献[1]至[7]。本文在这里提出的是一种基于用户角色和工作流程的公文传送机制,即:在公文文件数据库中设定各类文件的存取控制权限矩阵(如插入权、删除权、更新权、查询权等),分区管理文件,并将文件处理者分成各种角色来管理,再将公文进度划分为数个阶段作时间性管理。这里提出的安全模型是以用户角色为主体,机密文件为受体,角色的划分是以用户处理文件的方式为依据。受体不是以文件、关系表或记录为授权单位,而是以更细层次的单元(cell)为单位。委任授权的关系上,是由系统管理员规划文件类别,定义各类别的数据结构,并以存取矩阵设定主体对受体的存取权限,而将文件流程设定及用户角色的指定授权给文件制作者。本文还提出动态性的保护,即将文件处理的进度划分成若干个阶段,以事件驱动方式来调整主体对受体的存取权限。

二、公文流程设定

1. 基于用户角色的工作流程模型

工作流程图提供的目的是使系统能在适当时间(事件触发时)处理适当的工作。工作流程图[8][9]主要包含工作(work)与事件(event)。工作为指定的程序模组,事件则是用以驱动程序模组。为适应计算机协同工作(CSCW)的应用,本系统采用以基于用户角色的工作流程模型,使群组可以在客户机/服务器网络结构下,在适当时间扮演某种角色,并执行该角色应处理的工作。以公文管理系统而言,是让群组可以在适当时间以指定的角色处理公文,并将公文依序传递给下一位指定接收者。

2. 公文系统的文件流程

单位与单位间的文件传送是以网络数据交换技术实现,而单位内部的公文转移则须靠文件流程管理的技术,尤其当单位内部组织庞大时,文件流程管理就更显得重要了。若甲单位(如教务处)为发文端,乙单位(如某院系)为收文端,甲单位发文给乙单位前,须在单位内完成公文制作流程,从创稿、拟办、批示、公文定稿、组文件、发文以及文件归档,整个过程中可设定流程来进行公文转移。乙单位收文后,在单位内部进行公文处理,以执行甲单位交办业务,从收文、分文交办、拟办、批示决策、内部发文以及文件归档,整个循环过程中也可经由流程设定达到公文转移的目的。本系统的文件流程设定方式如下:

文件制作者输入数据并存为电子文件后，欲将此电子文件传递给其他用户时，可以设定文件的工作流程，工作流程图描述下列事项：

(1)接收者　文件必须依序传递给哪些用户。

(2)角色　各用户是以何种角色(身分)来处理接收到的文件。

(3)工作　为用户扮演该角色所应执行的程序。

(4)事件　改变工作流程的触发事件。

在公文制作传送过程中，为使公文制作呈现弹性，常需要动态增加拟办者，公文往往在传送的过程中才决定是否增加拟办部门或呈送最高领导批示。因此，若将文件流程的设定工作全权委托文件撰稿者，则缺乏弹性并且不实际。本机制允许制作群中每一位文件制作者附加流程。即撰稿者可以只在流程图中设定下一位接收者，再由下一位接收者完成后续流程。

为了避免动态变更可能引起泄密，文件转移时必须自动报告控制中心进行作业跟踪记录(Audit trailing)，由此系统不但可以掌握公文流程使得承办公文者可以正确得知公文流向外，还可记录泄密行为的责任归属。

3. 工作流程与存取控制的关系

本系统对于文件的安全管理采用两段式(two phase)策略，第一阶段以工作流程图确定用户角色、应执行的工作和文件流程。工作可视为程序或交易(transaction)，如撰稿、拟办等，因用户执行交易时须存取群体共用资源，如公文，故对于资源必须作进一步存取控制。第二阶段则是以下面所述的阶段性安全存取控制模型加以保护。

三、存取控制

存取控制是实现公文传送安全的主要手段，用户存取权限受到三种因素影响，即用户角色、受保护文件的范围及公文进度。

1. 分角色保护

每一用户对于文件的处理方式各有不同，因此产生各类型的角色，有必要针对不同角色设置不同的存取权限。一般来说。公文传送过程中包含五种角色：撰稿者、拟办签署者、批示签署者、归档者、历史文件查询者。

2. 最小实体单元的保护

指文件受系统保护的范畴，若以关系数据库结构加以剖析，可分为表格级(table level)、记录级(record level)、字段级(field level)、单元级(cell level)四种。

表格级：一份公文文件视为一笔记录，将相同结构的公文文件存在同一表格内。现行的关系数据库管理系统，如 sybase，oracle，informix 的文件存取控制均以表格对象为保护的最小实体单元。其授权给用户对某表格进行改动作业的语法如下：

Grant Insert | Delete | Update | Select

on Table_name

to User_id

记录级：一份文件为一笔记录，即授权于用户对整份文件作改动处理。对于公文而言，此种范畴仍太大，因一份公文必须加以分区保护，如撰稿者只能在基本数据区操作，而不得在签署区内操作；拟办者只能在拟办签署区签拟意见，而不得在其他区进行改动作业。

字段级：即授权于用户对相同类型文件的某字段作改动处理。例如将公文稿的拟办签署栏授权给用户甲，则用户甲可以对所有公文稿的签署栏加签意见。但实际应用上，用户甲应只在特定文件上具有拟办者角色，而非所有此类公文文件。

单元级：即授权于用户对特定文件的特定字段作改动处理。本系统是针对此种范畴层次作保护，因为只有此种范畴层次方能满足公文系统需求。

本机制以公文文件为处理对象的，公文的各项处理应符合各单位公文处理规范。此处将公文划分为四个区，每一区由一个以上的字段组成，这四区为基本数据区、拟办签署区、批示签署区和附件数据区。

3. 阶段性保护

文件的处理进度可划分为数个阶段，并且每一个不同角色的处理者只能在指定阶段处理文件，所以文件处于不同阶段也应设定不同的存取权限。本机制包含五阶段，即撰稿阶段、拟办阶段、决策阶段、归档阶段和文件查询阶段。

四、系统结构建立与运作

1. 存取参数

用户存取文件时，必须指明四个数据项。

(1)用户代号(User_id)：每一用户的文件存取权限储存在服务器端系统数据区内。系统分配给每一用户一个人公文数据库，并以用户代号为此数据库名称。故用户代号不仅表明欲存取的数据库外，也提供系统作身份辨识以核对存取权之用。

(2)公文文件代号(Doc_No)：指明欲存取的文件。

(3)文件区域(Block)：指明用户欲存取的区域。

(4)操作(operation)：即增、删、改、查四种基本操作。

2. 控制机制

本控制主要包含两种机制，即存取权检测器(access right checker)与文件存取通道(document access channel)。当用户发出用户代号等四个数据以存取数据库内的文件时，首先由存取权检测器检查用户存取文件时有无违背用户应有的权限，若违背则禁止用户存取该文件；如果用户未违背应有的权限，则允许用户存取该文件，由存取权检测器发出的控制信号允许还是禁止(permitted/forbidden)存取该文件，然后根据 User_id 由文件存取通道切换至对应的个人公文数据库以存取该文件。

3. 存取权控制器

存取权检测器包含三个控制表、一个阶段转换控制器(STD engine)及一个比较器。此

三个控制表为用户角色控制表(role profile)、存取权控制表(access right profile)及公文阶段控制表(stage profile)。分别描述如下：

用户角色控制表记录用户处理某文件时所应扮演的角色。其包含三个字段：文件代号(Doc_No)、用户代号(User_id)和角色(role)。此控制表的内容由撰稿者建立基本数据后，于设定工作流程的过程中加以指定。

存取权控制表记录角色对各公文区域在不同阶段下的存取权限。此控制表包含(block，role，stage，access right)四个数据字段，block，role，stage 三个数据字段的定义域如前述，access right 的可能值为(insert，delete，update，search)。此控制表由系统管理员负责建立与维护，系统管理员依据公文处理的安全需求加以建立。

公文阶段控制表记录各公文目前进行至哪一个阶段。当用户完成目前阶段的作业时，公文即进入下一个阶段。本系统用事件驱动(event driven)方式来维护此控制表，当用户完成目前阶段的作业时，便产生一事件以触发阶段转换控制器，阶段转换控制器依据内部的阶段转移图来改变公文的阶段。系统管理员依据公文处理的安全需求设定阶段转移图之后，阶段转换控制器便依此转移图来自动维护此控制表。

存取权检测器的工作如下：

(1)查询用户角色：对角色控制表输入用户代号、文件代号，经对比后取出用户对此文件具有的角色。

(2)查询被存取文件目前的阶段：对阶段控制表输入文件代号，经对比后取出该文件目前正进行的阶段。

(3)查询用户于目前阶段对该文件区域具有何种权限：对存取权控制表输入用户角色、文件目前阶段、欲存取的区域，经对比后取出用户应具有的权限。

(4)存取权对比：比较器核对用户对文件的操作(operation)有无超出使用者应有的权限，若超出权限，则 permitted/forbidden 控制信号为假(false)，反之为真(true)。

4. 效率与多重角色冲突问题

上述的存取控制模型，系统在存取文件前须经过严格的存取权检测，故系统的效率将会减低。为了改善效率，控制表可以用高速缓冲(cache)来加速检测速度。也可用软件加速法，即建立数据库存储程序(stored procedure)来处理检测作业，此方法改善的效果虽较差，但开发成本较低。

当群组制作同一份文件时，有时会发生多重角色冲突现象，即同一用户身兼数个角色(如公文撰稿者与归档者)。当具有多重角色的用户存取文件时，系统便进行存取权检测，在角色控制表内查询到该用户具有两个以上的角色，因在每一个时间点，任何用户只能扮演一个角色，此时系统是按照公文进度来判断用户角色，即系统会进一步查询阶段控制表，找出文件目前处于何种阶段，再挑选出最适当的角色。

五、结　论

本系统所提出的模型，可以做为面向公文处理和传送的基本模型。针对群组共同制作文件、角色分工和文件转移，本文提出基于用户角色的工作流程模型，使群组可以依照角

色来划分工作，此模型以事件驱动方式来维持工作流程的运转，系统可以有效并正确地依照流程顺序来进行文件转移。另外，针对公文文件传送因角色性错误、范围性错误或阶段性错误所造成的各种人为问题，本文提出阶段性存取权控制模型予以解决。

参 考 文 献

[1] G. S. Graham and P. L. Denning, Protection-principles and practice, Proc. Spring Joint Computer Conf., 40, AFLIPS Press, Montvale, Nj, pp. 417-429, 1972

[2] Denning et al., Views for multilevel database security, Proc. IEEE Symposium on Security and Privacy, 1986

[3] T. F. Lunt, Security in database systems: a research perspective, Computer Security, 11(1) pp. 41-56, 1992

[4] K. Dittrich, M. Hartig and H. Pfefferle, Discretionary access control in structurally object-oriented systems, Database Security, III: Status and Prospects, North-Holland, 1989

[5] T. C. Ting, A user-role based data security approach, Database Security: Status and Prospects, North-Holland, 1987

[6] F. H. Lochovsky and C. C. Woo, Role-based security in database management systems, Database Security: Status and Prospects, North-Holland, 1987

[7] N. R. Jensen, System security officer functions in the AI secure DBMS, Database Security II: Status and Prospects, North-Holland, 1988

[8] T. A. May, Know your work-flow tools, Byte, 19(7), pp. 103-108, 1994

[9] Meichun Hsu and Mike Howard, Work-flow and legacy system, Byte, 19(7), pp. 103-108, 1994

A user-role-based and phase control security system in documents management over Intranet

Chen Qingzhang

(Zhejiang University of Technology

No. 6 District, Zhaohui Xincun, Hangzhou, Zhejiang 310032, P.R. China)

Abstract The paper introduces a user-role-based and phases control security approach which applies to documents management over Intranet.

Keywords document-intranet, security system, access control, phase protection

OpenWeb 的安全机制

李春华　　周兴铭

（国防科学技术大学计算机学院，长沙，410073　Email：chunhuali@263.net　xmzhou@nudt.edu.cn）

摘　要　OpenWeb 是我们实现的一种基于 Web 的计算平台。它采用 Java 和 Cookie 来实现身份验证、访问控制等安全机制，使得用户可以利用普通的 Web 浏览器就可以登录到 Unix 主机，交互式地提交计算任务并浏览计算结果。

关键词　Web 安全　身份验证　访问控制　Web 计算　Java Cookie

一、前　言

Internet 的爆炸性增长，使得 World Wide Web 在商业、娱乐、教育、出版等领域不断发展，甚至进入高性能计算领域。WWW 和高性能计算相结合，使得用户可以通过互联网方便地访问 MPP，SMP，NOW 等计算资源：利用浏览器提交计算任务并浏览计算结果。这样做的最大优点是利用了浏览器的通用性和易用性，然而对于传统的 Web 服务器＋CGI（或 ISAPI，NSAPI 等）模式而言，由于缺乏交互性和实时性，它不能很好地支持 Web 计算。目前，Web 计算已成为一个研究热点。OpenWeb 是我们实现的一种 Web 计算平台，本文主要介绍它所采取的一些安全措施。

1. OpenWeb 简介

OpenWeb 结构如图 1 所示，它由 OpenWeb client 和 OpenWeb server 组成。OpenWeb client 是运行于普通浏览器中的 Java applet，包含 WebTerm client 和 IWP（Interactive Web Protocol）server 两个功能模块。OpenWeb server 是我们设计的一种特殊的 Web 服务器，它运行于 Unix操作系统，与一般 Web server 的主要区别是增加了 WebTerm server 和 IWP client。在 OpenWeb 环境下，用户通过运行于浏览器中的 WebTerm client 按 telnet 协议登录到 WebTerm server，WebTerm server 生成子进程执行一 Unix shell 作为用户的根

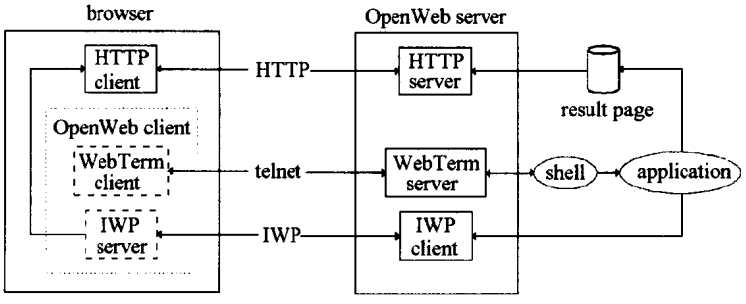

图 1　OpenWeb 结构

进程，用户通过此 shell 和服务器操作系统交互，在 shell 提示符下输入命令运行应用程序。在应用程序的执行过程中，它既可以直接向 WebTerm client 输出文本信息，如同输出到普通的仿真终端一样，也可以根据需要利用 IWP 协议向用户的浏览器输出超文本信息。IWP 协议是我们设计的一种交互式 Web 协议，它使 Web 服务器可以主动地和浏览器交互，例如向浏览器推送页面。当应用程序需要向浏览器窗口输出超文本信息时，它调用 OpenWeb server 提供的 API 通过 IWP client 发 IWP 请求报文给 IWP server，然后由 IWP server 通知浏览器取回相应的页面。

2. OpenWeb 的安全问题

WebTerm client 登录到 WebTerm server 时，按 telnet 协议，用户名和口令是明文传输的，很容易被第三方窃取。因此必须提供一种更为安全的方法。其次，用户登录成功后，浏览器发出的 HTTP 请求应能够标识出用户的登录身份，以便于对 HTTP 请求进行访问控制。

本文剩余部分内容安排如下：首先在第二节介绍 OpenWeb 安全机制的总体结构，接着在第三节描述其工作流程，第四节举两个应用 OpenWeb 的例子，最后在第五节给出总结并指出存在的问题。

二、OpenWeb 的安全机制

1. 总体结构

OpenWeb 安全机制的总体结构如图 2 所示，当用户用浏览器访问登录页面时，嵌入在登录页面中的 Java applet 即 OpenWeb client 开始运行，它接收用户输入的用户名和口令，然后用 OpenWeb server 的公钥加密并传给 OpenWeb server，OpenWeb server 用自己的私钥解密后得到用户名和口令，然后利用它们验证用户是否为操作系统的合法用户。如果不是系统的合法用户则登录失败，OpenWeb client 提示用户继续输入用户名和口令。否则登录成功并由 OpenWeb server 生成一进程运行/etc/passwd 文件中指定的 shell，如 C shell，Korn shell，Bourne shell 等。同时 OpenWeb server 生成该用户的身份证书，并通过 Cookie 传给浏览器，于是当浏览器再次向 OpenWeb server 发出 HTTP 请求时，请求中就

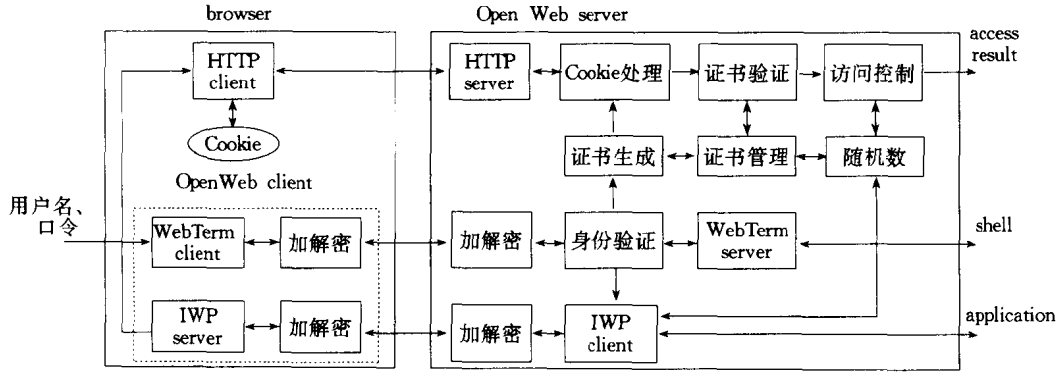

图2 OpenWeb的安全机制

包含着 Cookie，OpenWeb server 取出 Cookie 中的身份证书进行验证，确认用户身份，然后再进行访问控制，如果对相应的页面具有访问权限则从磁盘上读取并返回给浏览器，否则返回错误信息。

2. Cookie 和用户身份

当 Web 服务器响应浏览器的 HTTP 请求时，它可以把一段额外的信息包含在 HTTP 应答报文头部。这样浏览器接收到应答报文后，可以把这段信息存储起来，当它再次访问上述 Web 服务器时，就把这段信息包含在 HTTP 请求报文头部返还给服务器。这段额外的信息就称为 Cookie[2]。Cookie 机制在一定程度上克服了 HTTP 协议无状态的缺点。通过这种机制，Web 服务器可以根据需要存储某些信息在 Cookie 中，以此来记录浏览器的访问状态。因此，在 OpenWeb 中，我们利用 Cookie 来传递身份证书以便确认 HTTP 请求的用户身份。

为了在 HTTP 应答报文中包含 Cookie，可在报文头部插入 Set-Cookie 语句，其语法为：

Set-Cookie: NAME = VALUE; expires = DATE; path = PATH; domain = DOMAIN; secure

其中 NAME 为 Cookie 名，VALUE 为 Cookie 值，其余参数意义详见[2]。为了传递用户身份，OpenWeb server 把 NAME 置为用户名，VALUE 置为用户身份证书。用户身份证书包含用户 ID 及一随机数并用 MD5 加密。浏览器接收到 HTTP 应答报文后，将存储该 Cookie，当它再次访问 OpenWeb server 时，把 Cookie：NAME＝VALUE 包含在 HTTP 请求报文头部传给 OpenWeb server。这样 OpenWeb server 接收到 Cookie 后就可以从中取出用户名和身份证书，然后和自己存储的身份证书进行对比，从而确认 HTTP 请求的用户身份。

3. 随机数访问控制

对于应用程序通过 IWP 协议主动向浏览器推送的页面，还可以增加随机数访问控制。当应用程序产生一页面并通过 OpenWeb server 给 OpenWeb client 发消息时，OpenWeb server 产生一随机数并把它与该页面建立映射，然后把该随机数和页面 URL（如 http://openweb.nudt.edu.cn/result.html? random number）一块传递给 IWP client。IWP client 通过以下两条语句来通知浏览器去取相应的页面：

```
URL newUrl = new URL("http://openweb.nudt.edu.cn/result.html ? random number");
getAppletContext().showDocument(newUrl);
```

这样浏览器发出的 HTTP 报文头部请求行中就包含了随机数，如对上述 URL 有：

```
GET /result.html? random number HTTP/1.0
```

于是 OpenWeb server 在进行证书验证和操作系统文件读写权限的访问控制之后，还要进行随机数的访问控制，如果 HTTP 报文头请求行中的随机数和 OpenWeb server 自己所保存的一致，则把相应的页面返回给浏览器。

三、工作流程

OpenWeb 的工作流程如图 3 所示，包括以下步骤：

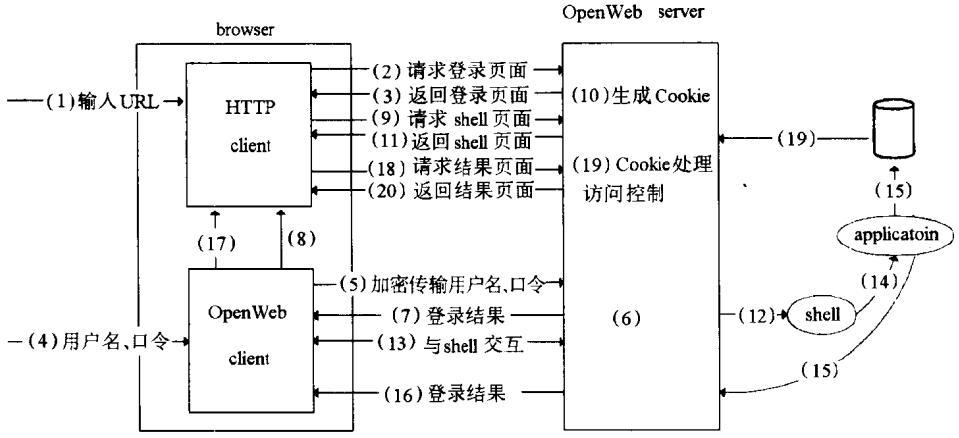

图 3　工作流程

(1) 用户在浏览器地址栏中输入 OpenWeb server 的 URL，例如：
http://openweb.nudt.edu.cn/login.html；

(2) 浏览器向 URL 所标识的 OpenWeb server 发送 HTTP 请求报文，要求访问登录页面 login.html；

(3) OpenWeb server 接收请求报文后通过 HTTP 应答返回登录页面 login.html，此页面包含 OpenWeb Client 的 Java applet 字节代码 OpenWebClient.class，因此浏览器接着下载 OpenWebClient.class 并运行它。

(4) OpenWeb client 运行后提示用户输入主机(例如 openweb.nudt.edu.cn)的操作系统帐号。此时用户可输入用户名和口令；

(5) OpenWeb client 对用户名和口令进行加密并传给 OpenWeb server；

(6) OpenWeb server 解密得到用户名和口令，进行身份验证，如果是合法的系统用户则登录成功并生成身份证书；

(7) OpenWeb server 向 OpenWeb client 返回身份验证结果；

(8) 如果登录失败，转向(4)，OpenWeb client 提示用户继续输入用户名和口令。否则 OpenWeb client 通知浏览器取 shell 页面 WebTerm.html；

(9) 浏览器向 OpenWeb server 发送 HTTP 请求取 shell 页面 WebTerm.html；

(10) OpenWeb server 生成 Cookie 并把身份证书包含在 Cookie 中；

(11) OpenWeb server 通过 HTTP 应答返回 WebTerm.html，应答报文头部包含着 Cookie，浏览器接收应答报文后显示 WebTerm.html 页面并存储 Cookie；

(12) OpenWeb server 为登录成功的用户生成子进程并运行 shell，然后把子进程的属主改为该用户；

(13) 用户和 shell 交互；