研究报告

基于 JavaBeans 与安全 Cookie 的 Web 应用安全中间件

蔡 准，孔凡玉，李大兴
（山东大学东校区网络信息安全研究所，济南 250100）

[摘要] 对 Web 应用中涉及信息安全的操作进行了讨论，设计了中间件层次构架，并在此基础上使用了 JavaBeans 与安全 Cookie 技术实现了安全中间件，从而通过硬件设备提高 Web 应用的效率及降低了开发难度。最后对基于 PMI 系统，通过 RBAC 对权限进行控制的扩展进行了讨论。

[关键词] JavaBeans；安全 Cookie；中间件；PMI；基于角色访问控制
[中图分类号] TP309 [文献标识码] A [文章编号] 1009－1742（2005）03－0074－04

1 开发背景

随着因特网的普及程度越来越高，电子商务、电子政务等 Web 应用已经迅猛地发展起来。由于各种 Web 应用是在 Internet 等开放网络上进行，因此信息传输的安全性是 Web 应用要解决的首要问题。

Web 应用中的安全性主要包括数据的机密性（confidentiality）、完整性（integrity）以及真实性（authentication）。数据的机密性，是指保证在开放的网络中传输的数据不被其他人非法窃取、窥探、篡改和破坏，这就要求对明文数据进行加密。数据的完整性和真实性是指要保证通信双方完成传送后，接收方获得的数据要完整，未被更改，并且发送方不可抵赖。实现这方面需要依靠公钥基础设施（PKI，public-key infrastructure），其核心就是证书和密钥的管理。

至于 Web 应用中其他的安全方面：物理网络安全、安全策略、访问控制等（比如防火墙等设施）不在此讨论。

中间件（middleware）是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源。中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。中间件是基础软件的一大类，属于可复用软件的范畴。顾名思义，中间件处于操作系统软件与用户应用软件的中间。中间件在应用软件的下层，但在操作系统、网络和数据库之上，总的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户灵活、高效地开发和集成复杂的应用软件。

世界著名的咨询机构 Standish Group 在一份研究报告中归纳了中间件的十大优越性[1]：缩短应用的开发周期；节约应用的开发成本；减少系统初期的建设成本；降低应用开发的失败率；保护已有的投资；简化应用集成；减少维护费用；提高应用的开发质量；保证技术进步的连续性；增强应用的生命力。

具体说，中间件屏蔽了底层操作系统的复杂性，使程序开发人员面对一个简单而统一的开发环境，减少程序设计的复杂性，将注意力集中在自己的业务上，不必再为程序在不同系统软件上的移植而重复工作，从而大大减少了技术上负担。

中间件不仅使应用系统开发简便、开发周期缩短，也减少了系统的维护、运行和管理的工作量，

[收稿日期] 2004－04－22；修回日期 2004－06－04
[基金项目] “九七三”国家安全重大基础研究资助项目（51214－04004）；国家自然科学基金资助项目（69901005）
[作者简介] 蔡 准（1978－），男，山东济南市人，山东大学网络信息安全研究所博士研究生

还减少了计算机总体费用的投入。Standish 的调查报告显示[1]，由于采用了中间件技术，应用系统的总建设费用可以减少 50 %左右。在网络经济大发展、电子商务大发展的今天，从中间件获得利益的不只是 IT 厂商，IT 用户同样是赢家，并且是更有把握的赢家。

其次，中间件作为新层次的基础软件，其重要作用是将不同时期、在不同操作系统上开发应用软件集成起来，彼此像一个天衣无缝的整体协调工作，这是操作系统、数据库管理系统本身做不到的。中间件的这一作用，使得在技术不断发展之后，以往在应用软件上的劳动成果仍然物有所用，节约了大量的人力、财力投入。

笔者建立了一个简单的三层构架，在客户端、服务器和底层的安全操作之间插入了中间件层次，从而满足 Web 应用中对信息安全方面的要求，并通过统一的接口，减轻和简化了开发 Web 应用平台时在不同系统、不同客户端软件上的重复开发工作。

2　结构框架

中间件基于三层构架，它的逻辑框图见图 1。在三层结构中，Web 应用的开发者不需要考虑安全操作的底层细节，以及操作系统，客户端软件之间的差别。只需要在中间件的上层直接开发业务流程，中间件负责完成所需要的相应安全操作。

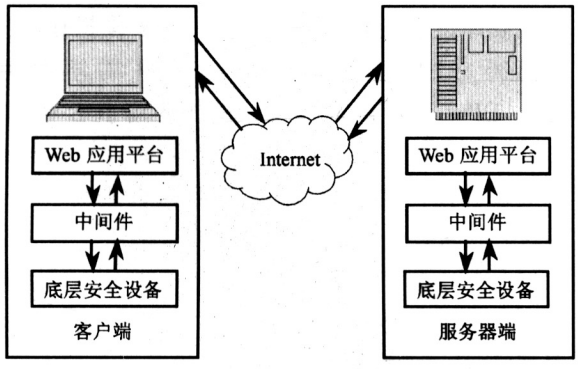

图 1　中间件逻辑构架

Fig.1　Middleware logic diagram

3　功能说明

Web 应用中客户端大多数使用浏览器，所提供的中间件通过对敏感数据进行加密和签名，从而确保数据的机密性、完整性、真实性。中间件屏蔽了在安全性操作中，各软件、各硬件、各平台的差异，通过统一的 API 提供给开发者，简化开发过程。

在客户端可选用 IC 卡、USB Key、加密卡等硬件设备，服务器端为了保证系统的性能不出现瓶颈，应使用高端的加密服务器等。

中间件使用 PKI 系统中 CA 所签发的 X.509[2] 证书，通过硬件设备在通信双方之间进行身份认证，验证客户端和服务器端证书的有效性、合法性。对于敏感数据进行加密，在网络中仅传输密文，保证敏感数据的机密性。对需要加密的数据运用散列函数并且签名，从而确保数据的完整性。

中间件应能够实现对不同浏览器、不同操作系统、不同设备接口标准的兼容。例如，在客户端可以使用 MS CryptoAPI[3]，PKCS # 11[4]，JCE (Java Cryptography Extension)[5]等接口，可以使用 IE，Netscape 等浏览器。由于服务器端的操作系统平台比较复杂，可能需要支持 Windows，Unix，Linux 等等，所以为了便于在多平台之间实现移植，服务器端以纯 Java 接口形式提供为宜（笔者推荐使用 JCE 接口）。

在客户端，主要讨论是以 MS CryptoAPI 接口为重点，因为这种接口的调用过程比较复杂，其他可以类似的实现。在服务器端，主要讨论使用 Java 语言的 JavaBeans 形式的实现。

在客户身份的认证方面，使用了安全 Cookie 技术，由服务器对客户的 Cookie 文件进行了签名，通过验证签名，以及验证 X.509 证书的有效性，可以确认客户的合法身份。

4　功能流程

设计的中间件主要模块包括安全 Cookie 登录、安全 Cookie 验证、登出、消息加密、消息签名、消息签名并加密等。

4.1　安全 Cookie 登录（Login）

客户端申请登录网址，服务器生成一个随机数发送到客户端，客户端 Applet 使用 JNI（Java Native Interface）调用动态库，实现对 CryptoAPI 的调用，将服务器发送的随机数签名，通过表单将签名的结果以及 Applet 获得的客户端证书提交给服务器（以 PKCS # 7[6]的格式提交）。由服务器端 JSP 脚本调用 JavaBeans，验证客户端证书的正确性以及证书时间、证书吊销列表（CRL）等，如果

验证通过，服务器JSP脚本中通过JavaBeans，使用客户端证书验证客户端对随机数的签名，如果验证通过，则验证客户身份的工作完成，可以确定为合法用户，将用户信息、IP地址等写入Cookie文件，并在其中写入服务器对Cookie信息的签名值，将Cookie写入客户端，重定向到登录成功页面。如果验证证书或者验证签名时出错则登陆失败。只有在客户登录成功以后才可以进行消息的签名、加密操作。

4.2 安全Cookie验证（Verify）

服务器获得用户的安全Cookie信息。验证Cookie生命期是否已经到达，验证IP地址是否与客户端相符，并且验证服务器对客户端Cookie的签名信息，如果验证通过，说明用户已经登录过，是合法的用户，允许用户访问资源，否则直接导向至安全Cookie登录页面。

4.3 登出（Logout）

由服务器删除客户端安全Cookie，保证在退出后登录失效。如果遇到意外退出，在Cookie生命值长度到达以后自动失效。

4.4 信息签名（SignMessage）

客户端向服务器申请签名网页，服务器脚本调用安全Cookie验证模块，验证客户端Cookie合法性。如果有效，导向到信息签名表单页，否则导向到登录页面。用户提交表单，Applet通过类文件使用JNI调用动态库，实现CryptoAPI调用，使用用户的证书对信息签名后提交给服务器。服务器脚本通过JavaBeans由LDAP获得用户证书对签名验证，验证通过后，导向到成功页面，否则到错误页面。

4.5 信息加密（EncryptMessage）

客户端向服务器申请加密网页，服务器脚本调用安全Cookie验证模块验证客户端Cookie合法性。如果有效，导向到信息加密表单页，否则导向到登录页面。用户提交表单，Applet通过类文件使用JNI调用动态库，实现CryptoAPI调用，使用对称密钥对信息加密，并用服务器公钥封装对称密钥为数字信封格式后提交给服务器。服务器脚本通过JavaBeans用自己的私钥解密对称密钥，使用获得的密钥对信息解密，解密成功后，导向到成功页面，否则到错误页面。

4.6 信息签名，加密（SignAndEncryptMessage）

客户端向服务器申请签名、加密网页，服务器脚本调用安全Cookie验证模块验证客户端Cookie合法性。如果有效，导向到信息签名、加密表单页，否则导向到登录页面。用户提交表单，Applet通过类文件使用JNI调用动态库，实现CryptoAPI调用，使用自己的证书对信息签名、使用对称密钥对信息加密，并用服务器公钥封装对称密钥到数字信封格式后提交给服务器。服务器脚本通过JavaBeans由LDAP获得用户证书验证用户的签名，使用自己的私钥解密对称密钥，使用获得的对称密钥对信息解密，解密成功后，导向到成功页面，否则到错误页面。

5 如何实现接口调用

在中间件的实现中最关键的是接口的调用，客户端可以使用CryptoAPI，PKCS#11，JCE等接口，服务器端一般应使用JCE接口。

5.1 客户端

由于在客户端使用了MS CryptoAPI接口，这个接口是C语言的动态库。在Java环境中调用C语言动态库需要使用Sun公司提供的JNI方式[7]。

Java 2平台中，为了安全起见对Applet的权限进行了严格限制，在Applet中是无法直接读入客户端系统中动态库的，可以在另外一个类“B”的文件中实现JNI的调用，然后在Applet中声明一个“B”类的静态对象，从而实现在Applet中调用“B”类中的静态方法。通过这种方式，成功的对C语言动态库进行调用，完成所需要的安全操作。

值得注意的是，如果在客户端不使用CryptoAPI这类C语言接口，例如使用JCE接口，这样就不必使用JNI的调用方式，同时客户端的跨平台等兼容性得到很大提高，而且可以避免在JNI调用时，Java平台对代码失控的安全隐患。这里采用CryptoAPI接口只是作为一个例子，并不代表推荐使用，在中间件的实现过程中，完美的解决方案是使用纯Java来编写。

5.2 服务器端

服务器端应尽量使用JCE接口的形式，因为这样可以保证服务器端的良好兼容性，很容易就可以移植到多种操作系统，支持IIS，Apache，Tomcat等多种Web服务器。

为了在JSP脚本中使用JCE接口，需要实现JavaBeans的形式。在JavaBeans的类中使用JCE接口完成安全的相关操作（签名、加密等）后，就可

以在 JSP 中使用 JavaBeans 来工作了，JSP 中使用 JavaBeans 的语句如下：

〈jsp：useBean id = "Cryptography" scope = "session" class = "test. Bean" /〉

其中 id 表示 Bean 在 JSP 中使用时的名称，scope 表示 Bean 的生存期，class 表示编译后的 JavaBeans 类。

JavaBeans 的具体编写格式见文献[8]。

6　对 Applet 的一点补充

文中客户端的安全操作使用了 Java applet，通过 Applet 调用动态库文件，使用 JNI 实现对 CryptoAPI 接口的调用。由于使用了本地资源，所以 Applet 必须获得足够的权限，而在 Java 2 平台中，一般的 Applet 是不允许操作本地资源的。

为了获得访问资源的权限，需要对自己编写的 Applet 进行签名，这样 Applet 才能在浏览器中正常使用。

在 Java 2 平台中已经提供了对 Applet 进行签名的工具，将生成的 class 文件打包为 JAR 的格式，然后生成用来签名的密钥库（keystore）和密钥（key），这样就可以使用命令对 JAR 包进行签名，命令如下：

jarsigner-keystore A. keystore B. JAR

其中 A 代表密钥库名称，B 代表打包后的 JAR 文件包的名称。

7　与 PMI 结合通过 RBAC 实现权限控制

如果在 Web 应用中需要对权限进行控制，还可以结合 PMI 系统，使用基于角色的访问控制（RBAC）来控制访问资源的权限。

具体实现时可以在安全 Cookie 登录模块中，双向验证通过以后，由服务器从 LDAP 取得用户的属性证书，并且从中取得用户的角色信息添加到安全 Cookie。

在用户每次请求资源的时候服务器在验证了安全 Cookie 有效性以后，由 Cookie 中获得用户的角色信息，解析用户请求的 URL 获得用户请求的资源信息，使用用户的角色信息、策略文件信息，调用模块进行权限判断，决定是否允许用户进行相应资源的访问。

8　结语

随着市场竞争的加剧，以及 Web 应用发展的需要，开发者经常要进行业务流程的调整，越来越多的企业开始关注系统的伸缩性和可扩展性。而中间件可以支持跨平台操作，为不同操作系统上的应用软件集成提供方便，从而满足企业内部对系统伸缩性和可扩展性的要求。由于以上因素的作用，中间件已经成为企业应用的“新宠”，在银行、电信、金融等大规模关键事务领域中的整合各种异构平台、保证信息完整性等方面，表现出了超强的能力。

笔者就 Web 应用中间件中信息安全有关的操作进行了讨论，提供了实现方案，成功实现了屏蔽底层系统、设备差异的目的。通过硬件设备可以有效的大幅提高 Web 应用中的业务处理速度，从而提高系统本身的能力。另外由于硬件设备的安全性要远比软件高，设备中的私钥不可取出，保证了加密，签名数据更高级别的安全性。现在市场上的硬件设备一般都可提供 MS CryptoAPI，PKCS＃11，JCE 等接口标准，通过对文中实现过程进行扩展，很容易支持所有的这些标准，也很容易实现对跨平台、跨系统的要求。通过中间件的设计，对于不同的设备，不同的操作系统等提供了统一的形式，简化了 Web 应用的开发过程，大大降低了开发难度。文中的实现方式具有一定的理论及实用价值，并且已经在多个系统下测试通过。中间件是一个很广泛的概念，文中实现的只是涉及安全的操作，对于其他的功能没有进行过多讨论，还有很大的扩展余地，只有对其功能进一步扩充才能真正成为一个实用的系统。

参考文献

[1] Standish Group International. Middleware Handbook [M/OL]. Standish Group International, http: // www. marketresearch. com/product/display. asp? Productid = 719883, 2001 - 10 - 01

[2] R Housley, W Ford, W Polk, D Solo. Internet X. 509 Public Key Infrastructure, Certificate and CRL Profile [S]. RFC 2459, IETF, 1999 - 01

[3] Robert Coleridge. The cryptography API, or how to keep a secret [DB/OL]. Microsoft Developer Network Technology Group, MSDN Library, 1996 - 08 - 19

（下转第 83 页）

Treatment of Highly Weathered Rock Mass at the Abutment of Earth-Rock Dam in Qiaoqi Hydropower Project

Wang Shougen

(*Chengdu Hydroelectric Investgation & Design Institute of State Power Corporation, Chengdu* 610072, *China*)

[**Abstract**] The abutment is the part where the by-pass seepage and contact erosion often happen to make the dam not work normally, which is the key point during design and construction, so Design Specifications for Rolled Earth-Rock Fill Dams (SL274 - 2001) require excavating into the slightly weathered rock mass for the core wall and the reversed filter and the abutment. The spreading gradation earth-rock fill dam with crushed-stone-earth core wall is 123m high, and the highly weathering natural depth of the phyllite and sandstone at abutments on both bank is 14.6～66.4m. According to the Specification, the excavation quantity is enormous. It is practically excavated into the highly weathering zone, which can not meet with the Specification. The treatment scheme for the highly weathering rock mass on both abutments and grouting tests and grouting technics are introduced in detail. The feasibility to adopt consolidation grouting and seepage treatment measures is argued and the treatment measures and construction control standards are proposed.

[**Key words**] Qiaoqi Hydropower Station; earth-rock dam; treatment of abutment; highly weathered rock mass; consolidation grouting; curtain grouting

(cont. from p.77)

[4] RSA Laboratories. PKCS #11: Cryptographic Token Interface Standard [S/OL]. http://www.rsasecurity.com/rsalabs/node.asp?id=2133, 2001-11

[5] Sun Microsystems. Java™ Cryptography Extension (JCE) Reference Guide [S/OL]. Sun Microsystems, http://java.sun.com/j2se/1.5.0/docs/guide/security/jce/JCERefGuide.html, 2004-01-29

[6] RSA Laboratories. PKCS #7: Cryptographic Message Syntax Standard [S/OL]. http://www.rsasecurity.com/rsalabs/node.asp?id=2129, 1993-11-01

[7] Sheng Liang. Java™ Native Interface: Programmer's Guide and Specification [M]. Addison-Wesley Pub Co, 1st edition, 1999-06-10

[8] Greg Voss. Introducing Java Beans [C]. Sun Microsystems, http://java.sun.com/developer/onlineTraining/Beans/Beans1/, 1996-11

A Security Middleware for Web Applications Based on JavaBeans and Secure Cookie

Cai Zhun, Kong Fanyu, Li Daxing

(*Institute of Network Security, Shandong University, Jinan* 250100, *China*)

[**Abstract**] This paper discusses the secure operations in the web applications, and designs a framework of middleware. The paper implements the security middleware based on the "JavaBeans" and "Secure Cookie" technology. The middleware improves the efficiency of the web applications through hardware, and reduces the difficulty of development. In the end, the paper discusses the extension based on PMI system and RBAC to control the privilege.

[**Key words**] JavaBeans; Secure Cookie; middleware; PMI; RBAC