PKI 证书的撤销与验证

徐海琛，魏柏丛
（北京邮电大学 信息网络中心，北京 100876）

摘 要：随着电子商务公司越来越多地使用数字证书（由认证中心签发的电子身份证）来保证在线交易的安全性，这一行为引发了对另一种安全性的需要，即对数字证书的有效性进行验证。因此，CA 认证的一个重要操作就是验证用户的证书是否被撤销或者挂起。证书的撤销采用证书撤销列表，而用于验证证书状态的机制一般使用轻型目录存取协议或者在线证书状态协议。简要介绍了证书撤销列表以及基于轻型目录存取协议的目录服务机制，而重点讨论了基于在线证书状态协议的目录服务，并例举了一个在线证书状态协议的实际应用。

关键词：公钥基础设施；证书撤销列表；轻型目录存取协议；在线证书状态协议
中图法分类号：TP393 文献标识码：A 文章编号：1001-3695(2002)09-0064-04

The Revocation and Verification of PKI Certificates

XU Hai-chen, WEI Bo-cong
(*Information Network Center, Beijing University of Posts & Telecommunictions, Beijing* 100876, *China*)

Abstract: With more and more E-commerce business companies use digital certificates(a kind of electronical identification signed by Certificate Authority) to ensure the security of on line trade, it triggered another need for security, that is to verify the digital certificates' validity. One of the important procedures in Certificate Authority is to verify whether the user's certificate has been revoked or hold. We always use Certificate Revocation List to save the revoked certificates, and use Lightweight Directory Access Protocol or Online Certificate Status Protocol as verification mechanism. This thesis mainly focused on the Directory Service based on OCSP, but simply introduced the Directory Service based on LDAP, and used an OCSP implementation as an example.

Key words: Public Key Infrastructure(PKI); Certificate Revocation List; Lightweight Directory Access Protocol; Online Certificate Status Protocol

1 引言

随着 Internet 电子商务、网上银行业务以及政府上网工程的快速发展，利用信息安全技术来保障在开放的网络环境中传输敏感信息的问题越来越受到人们的广泛重视。在这一领域中，当今世界上采用的主流技术是公钥基础设施（Public Key Infrastructure，PKI）。

PKI 是一个利用现代密码学中的公钥密码技术，在开放的网络环境中提供数据加密以及数字签名服务的统一密钥管理平台。PKI 的基本机制是定义、建立身份认证以及授权证书，然后分发、交换证书，并在网络之间解释、管理这些证书。PKI 提供了访问控制服务、通信保密服务、完整性服务、认证服务、不可抵赖服务、会话保密服务、密钥恢复服务、时间戳服务等信息安全必须的功能，支持电子商务的参与者在网络环境下建立和维护平等的信任关系，保证网上在线交易的安全。总之，利用 PKI 技术来管理在开放网络环境中使用的公开密钥和数字证书，可以使互相看不见的用户通过证书信任链进行安全交流，从而就可以建立起一个相对安全、值得信赖的网络环境。

PKI 体系主要由三大组件构成，即认证中心、注册中心和目录服务。图 1 是一个 PKI 体系的结构简图。

其中，认证中心（Certificate Authority，CA）主要完成对证书的管理，具体包括：生成 RSA 密钥对和根证书；审查由 RA 提交的证书申请，并使用 CA 私钥对获准的证书申请签名；发布证书和证书撤销列表到 DS 的内部数据库中；颁发用户证书以及实现交叉认证等。由于 CA 采用数字签名技术，任何对证书内容的非法修改，都会被用户经使用 CA 的公钥进行验证而发现；因此，数字证书的合法性和数据的完整性是有保证的、可以信赖的，能够满足在开放的网络上发布敏感信息的要求。

注册中心（Registration Authority），RA 是终端用户 EE 和 CA 的接口，由 RA 获得的用户标志的准确性是 CA 颁发证书的基础。RA 的主要功能是审查证书申请者的身份。这项功能通常由人工完成，也可以由机器自动完成，但是 RA 必须具有身份检查机制，具体包括：验证申请者身份；批准合法的证书申请并提交 CA 签发；当用户发生变化时，向 CA 申请撤销其证书等。

目录服务（Directory Service，DS）的主要功能是维护和发布其内部数据库中由 CA 签发的证书和证书撤销列表（CRL）。DS 可采用基于 LDAP 协议的目录机制，用户定期从 LDAP 服务器上下载整个 CRL 到客户

端,然后在客户端执行证书的验证;或采用基于 OCSP 协议的目录机制,直接在 DS 服务器端实时查询证书的最新状态。

终端用户(End Entity,EE)为 PKI 体系中证书的具体使用者。

2　证书的撤销

出于安全的角度考虑,每一份证书都有其生存期。X.509v3 证书中有一个有效期限(validity)字段,规定了证书的签发日期(notBefore)和失效日期(notAfter),过期的证书必须作废。此外在有些情况下,如用户名字(subjectName)的更换、私钥(privateKey)被泄露或攻破、机构重组等等,尽管用户证书还没有过期,但是也必须撤销。被撤销证书的信息由 DS 对外发布。

在 X.509v3 标准中使用证书撤销列表(Certificate Revocation List)机制实现证书的撤销。CRL 是一个已经被撤销(Revoke)或挂起(Hold)的证书的列表。由于 CA 的签名可以验证 CRL 的真实性和完整性,所以 CRL 可以存储于网络上的任何节点。而且,在 CRL 中还包含一个颁发日期(thisUpdate),以及下一个 CRL 的颁发日期(nextUpdate)。由这两个日期信息,用户可以确定当前拥有的 CRL 是否是最新的,以及用于帮助管理 CRL 缓冲区:即在下一个 CRL 颁布之前,用户可以一直使用原来的 CRL 缓冲区。CRL 的数据结构如下:

```
CertificateRevokeList ::= SEQUENCE {
    tbsCertificate          TBSCertRevokeList,
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING
}

TBSCertRevokeList ::= SEQUENCE {
    version                 Version,
    signature               AlgorithmIdentifier,
    issuer                  Name,
    thisUpdate              Time,
    nextUpdate              Time,
    revokedCertificates     SEQUENCE OF SEQUENCE {
    userCertificates        CertificateSerialNumber,
    revocationDate          Time,
    crlEntryExtensions      Extensions OPTIONAL {
      crlReason             CRLReason,
      invalidityDate        Validity,
      certificateIssuer     Name
      }
    }
}
```

终端用户可根据 CRL 中是否包含待验证的证书来判断证书的有效性。该过程分为三个步骤:第一是获取相对应的 CRL;第二是校验 CRL 的数字签名是否有效;第三是检查待验证的证书是否在 CRL 中。

在网络的规模不大、证书的撤销率不是很高的情况下,采用 CRL 是一个十分有效、有较好伸缩性的办法。而且还可以进一步利用 X.509v3 证书中的 CRL 分发点(CRLDistributionPoints)扩展字段,指定适用于该证书的 CRL 应该从什么地方下载,从而允许 CA 将整个 CRL 空间划分为多个小的子空间。这样一来,即使证书的撤销率较高,各个 CRL 片段的大小也可以得到很好的控制。用户如果需要验证某特定证书的状态,即可大大地减少下载的数据量。但是 CRL 机制存在着固有的一些缺点。例如,当网络规模变大时,用户基础很大,证书机构会经常签发 CRL,导致 CRL 迅速增长,变得越来越大。减少更新时延和减少占用带宽总是矛盾的,由此带来 CRL 的不可靠(不能及时反映被撤销的证书)和可能成为瓶颈(每次验证证书链上的一个证书都必须下载相应的 CRL)。无论采取哪些改进措施,如 CRL 分发点、CRL 缓存、CRL 分组、Delta-CRL、最新撤销信息指针和重定向指针等,都不能从根本上解决问题。

3　证书状态的验证

3.1　基于 LDAP 协议的目录服务

LDAP(Lightweight Directory Access Protocol)并不是一个全新的协议,而是由 X.500 目录存取协议(Directory Access Protocol)经简化修改得来的,即 RFC2251。LDAP 协议基于面向连接的、可靠的 TCP 协议来实现。在信息安全领域中,LDAP 一般用于存取证书以及证书撤销列表(CRL)。当图 1 中的 DS 服务器运行 LDAP 协议时,则成为 LDA 服务器,如图 2 所示。

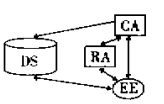

图 1　PKI 体系的结构图

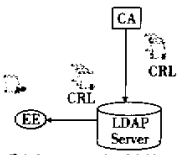

图 2　采用 LDAP 机制的 DS 服务

图 2 中,CA 将整个 CRL 签名后发布到 LDAP 服务器的内部数据库中;而用户对于证书状态的验证,则是通过 LDAP 服务器将整个 CRL 下载到本地的机器上,再根据下载的 CRL 进行查询。由于验证在用户本地进行,因而速度快,适用于大量突发性的验证请求。

LDAP 协议基本上由以下四个模块组成:

(1)信息模块(Information)。它定义了 LDAP 的数据结构。LDAP 的数据在逻辑上是以树状结构存储,即目录信息树(Directory Information Tree,DTT)。DIT 树由不同等级的节点组成,并按照国家(c)、组织(o)、部门(ou)、普通名称(cn)的分级方法组织存放信息。树中的每一个节点表示一个对象,即目录项(Entry);每一个 Entry 又包含若干个属性(Attribute);而每一个 Attribute 又是由一个类型(Type)以及若干个与之相关的值(Value)构成。

(2)命名模块(Naming)。它定义了在 LDAP 中应如何命名以保证每一个 Entry 的名字惟一。每一个 Entry 的一个或多个 Attribute 值构成其相对惟一名称(Relative Distinguished Name,RDN)。该 RDN 名称在 DIT 树中相应 Entry 的兄弟节点中是惟一的;由 Entry 到树根的若干个 RDN 的串联构成了该 Entry 的惟一名称(Distinguished Name)。因此,在 LDAP 中,数据的存取以 DIT 绝对路径的方式进行,但是绝对路径的地址却是由底向根增长,这与文件系统中由根向底增长的绝对路径方式恰好相反。

(3)功能模块(Function)。它定义了 LDAP 提供给用户的功能。其基本上可分为三类:①查询(Inter-

rogation)类,有 Search,Compare 两种功能;② 更新(Update)类,有 Add,Delete,Modify,Modify RDN 四种功能;③认证(Authentication)类,有 Bind,Unbind,Abandon 三种功能。

(4)安全模块(Security)。它定义了 LDAP 上的安全策略。普通用户除了可以查询数据以外,在未经过有关权力机构的认证并授予其相应权限的情况下,是不允许执行如 Add,Delete,Modify 等动作的。同样,该认证策略也以用户的证书为基础,不同等级的证书决定了证书主体访问数据的权限。

由于 CRL 带有 CA 的数字签名,因而发布 CRL 的 LDAP 服务器并不一定需要是可信的。同时,由于 LDAP 服务器可被镜像和备份,以及结合使用证书的 CRLDistributionPoints 字段,可组建 LDAP 服务器群,从而避免在网络通讯中形成瓶颈,从整体上提高系统的可靠性。

但是在 LDAP 技术中,由于 CRL 的发布是周期性的,所以有时并不能反映当前最新的证书状态。若发布周期较短,用户不得不每次都下载 CRL,占用了大量的时间和网络带宽 如果撤销频率很高,则 CRL 常常会变得很大,每次下载很大的 CRL,其中又有很多对特定用户无用的信息。因此,对于小型网络的使用者,由于并非随时面对大量的验证请求,下载 CRL 就成了一个很大的负担。

3.2 基于 OCSP 协议的目录服务

OCSP 协议正是针对上述 LDAP 的缺点而推出的在线证书状态协议(Online Certificate Status Protocol),即 RFC2560。尤其是在收到敏感证书,需要进行实时验证时,借助于 OCSP 即可了解到证书的最新状况。当图 1 中的 DS 服务器运行 OCSP 协议时,就成了 OCSP 应答器,如图 3 所示。

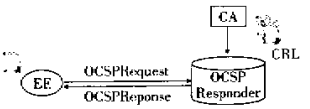

图 3 采用 OCSP 机制的 DS 服务

如图 3 所示,基于 OCSP 协议的 OCSP 应答器(OCSP Responder)可以为用户提供实时的证书状态查询服务。CA 将整个 CRL 签名后发布到 OCSP 应答器的内部数据库中。当用户需要验证证书的状态时,则向 OCSP 应答器提交一份查询服务请求信息(OCSPRequest),OCSP 应答器查询其内部 CRL 数据库,然后把当前最新的证书状态信息(OCSPResponse)以签名的形式响应给用户。以下是一个实际应用的以 RFC2560 为基础的 OCSP 查询流程,分为四个步骤:

(1)当客户需要验证其证书状况时,首先向 OCSP 应答器发送一个以用户密钥签名的证书状态查询请求 OCSPRequest。

①提取客户端的当前系统时间作为时间戳 RequestTime;

②产生顺序递增的序列号 serialNumber 和不重复的随机数 Nonce,以防止伪造和抵御重送攻击;

③按照待验证证书的顺序生成查询请求链表 requestList,该链表由一系列的查询请求 Request 组成;

④结合 requestTime、nonce 和 requestList 生成证书状态查询请求信息包 tbsRequest;

⑤用单向 Hash 函数计算生成 tbsRequest 的消息摘要 sendReqDigest;

⑥使用用户的 RSA 私钥对 sendReqDigest 加密,生成 tbsRequest 的数字签名 Signature,并将 Signature 附在 tbsRequest 的后面作为该数据包的校验码,形成证书状态查询请求消息 OCSPRequest。其数据结构如下:

```
OCSPRequest ::= SEQUENCE {
    tbsRequest              TBSRequest,
    signature               Signature
}
TBSRequest ::= SEQUENCE {
    version                 Version,
    serialNumber            INTEGER,
    subjectName             Name,
    requestTime             Time,
    nonce                   INTEGER,
    requestList             SEQUENCE OF Request,
    requestExtensions       Extensions OPTIONAL
}
Signature ::= SEQUENCE {
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING,
    certificates            SEQUENCE OF Certificate OPTIONAL
}
```

(2)当 OCSP 应答器收到客户的服务请求后,首先对接收到的用户证书状态查询请求信息 OCSPRequest 进行认证。

①使用用户的 RSA 公钥对收到的 OCSPRequest 中附在 tbsRequest 后面的数字签名 Signature 解密,还原出原消息摘要 sendReqDigest;

②使用与用户一致的单向 Hash 函数,再次对 tbsRequest 计算生成新的消息摘要 recvReqDigest;

③将两个消息摘要 sendReqDigest 和 recvReqDigest 进行比较,看是否完全相同;

④若相同,则接受该请求 否则,拒绝该请求。

(3)OCSP 应答器对于接受的查询请求在其内部 CRL 数据库中执行查询,确定证书状态,并将查询结果以 OCSP 应答器签名的证书状态查询响应 OCSPResponse 的形式反馈给用户。

①提取当前系统时间作为时间戳 responseTime;

②重用用户请求的序列号 serialNumber,并产生一个新的随机数 Nonce,用于防止伪造以及抵御重送攻击;

③根据用户的查询请求链表 requestList 依次在其内部的 CRL 数据库中执行查询,并将查询结果存储于查询响应链表 responseList 中;

④结合 responseTime,Nonce 和 responseList,生成证书状态查询响应信息包 tbsResponse;

⑤使用与用户一致的单向 Hash 函数计算生成 tbsResponse 的消息摘要 sendRepDigest;

⑥使用 OCSP 应答器的 RSA 私钥对 sendRepDigest 加密生成 tbsResponse 的数字签名 Signature,并将 Signature 附在 tbsResponse 的后面作为该数据包的校验码,构成证书状态查询请求消息 OCSPResponse。其数据结构如下:

```
OCSPResponse ::= SEQUENCE {
    tbsResponse                 TBSResponse,
    signature                   Signature
}
TBSResponse ::= SEQUENCE {
    version                     Version,
    serialNumber                INTEGER,
    responderName               Name,
    responseTime                Time,
    nonce                       INTEGER,
    responseList                SEQUENCE OF Response,
    responseExtensions          Extensions OPTIONAL
}
Signature ::= SEQUENCE {
    signatureAlgorithm          AlgorithmIdentifier,
    signatureValue              BIT STRING,
    certificates                SEQUENCE OF Certificate
}
```

(4)用户对 OCSP 应答器反馈的信息进行分析,确定待查证书的状态。该过程与步骤二类似,不再赘述。

若一切正常,OCSP 应答器则响应给用户查询的结果为三种状态之一:①Good,证书未被撤销;②Revoke,证书已被撤销或暂停使用;③Unknown,服务器内无此证书的信息。

若在查询过程中出现错误,则在 OCSP 中定义如下种错误:

- MalformedRequest　查询请求不合 OCSPRequest 的文法
- InternalError　OCSP 应答器内部故障
- TryLater　OCSP 应答器暂时无法提供服务,请稍后再试
- SignRequest　请求必须加数字签名
- Unauthorized　客户没有进行该查询的权限

本文的应用是在 RFC2560 标准的基础上对 OCSP 的查询/应答消息的数据结构进行了简化,既提高了证书验证的效率,又能满足我们的实际需要。

由于 OCSP 应答器总是实时获取信息,因此能够反映证书的真实状态。同时与 LDAP 相比,每一次证书验证需要处理的信息要少得多,相应地,客户端应用程序中需要处理的代码也相应少一些。但是,这种基于 OCSP 协议的方法要求 OCSP 应答器必须是一个可信的在线服务器(而发布 CRL 的 LDAP 服务器则不需要是可信的),并且能为每一个请求提供及时的响应。

4　结束语

综上所述,PKI 体系中的 DS 服务一般采用 CRL 作为其证书撤销库的数据结构,而采用 LDAP 或者 OCSP 作为 CRL 的发布协议。但是在实际应用中通常使用 LDAP 与 OCSP 并存的发布机制。用户可以利用网络相对空闲的时段,借助于 LDAP 协议从 DS 服务器下载整个 CRL 到客户端,执行本地查询;而在网络繁忙时段,则借助于 OCSP 协议在 DS 服务器端在线实时查询证书的状态,然后将查询结果反馈给用户。这样一来,既能满足突发性大量信息的查询请求,又能保证对敏感证书的实时验证,从而既保障了证书使用的安全,又相对平衡了网络负载,提高了工作效率。

对张晓军同学在此表示感谢。

参考文献:

[1] 陈彦学. 信息安全理论与实务[M]. 北京:中国铁道出版社, 2001. 124-151.
[2] 周武,冯登国. 联邦公钥基础设施(PKI)技术简介[J]. 密码与信息,1999,(3):24-62.
[3] Bruce Schneier B. 应用密码学——协议、算法与 C 源程序(第二版)[M]. 北京:机械工业出版社, 2000.118-131.
[4] Housley R,et al. Internet X.509 Public Key Infrastructure, Certificate and CRL Profile[S]. RFC2459,1999.
[5] Wahl M, Howes T, Kille S. Lightweight Directory Access Protocol(v3)[S]. RFC2251,1997.
[6] Myers M,et al. X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol—OCSP[S]. RFC2560, 1999.
[7] RSA Laboratories. PKCS#7:Cryptographic Message Syntax Standard[Z]. Version 1.5,1993.

作者简介:

徐海琛(1971-),男,硕士研究生,研究方向为网络安全;魏柏丛(1944-),男,高工,研究方向为计算机网络。

(上接第 63 页)

__global_sti()是开中断函数,它首先判断 CPU 是否正在处理 IRQ,如果没有,则释放全局 IRQ 锁,然后允许在此 CPU 上继续进行中断。而关中断函数__global_cli()的流程图如图 4 所示。

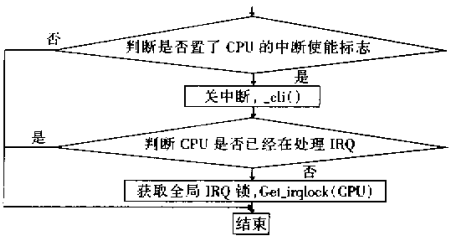

图 4　函数__global_cli()流程图

4　结束语

在本文截稿时 Linux 操作系统已经推出了其最新稳定版本 2.4.0 版。新版的 Linux 操作系统核心在高性能和多处理器计算机系统上的兼容性均优于 2.2 版的核心。举例来说,Linux2.0 只有一个单独的全局内核锁,所以每次只有一个 CPU 可以在内核里执行。Linux2.2 把这个全局内核锁使用的大部分地方都用更小的、子系统专用的锁来替代了。而 Linux2.4 继续深入,它把可能减小其作用域的锁都做了进一步的分割。比如 Linux2.4.0 版中每个等待队列就有一个锁,而不是所有的等待队列才有惟一的一个锁。相信随着 Linux 内核版本的不断更新,Linux 操作系统对 SMP 的支持也会日益进步。

收稿日期:2001-11-14;修返日期:2002-03-08

参考文献:

[1] [美]Scott Maxwell. Linux 内核源代码分析[M]. 冯锐,等. 北京:机械工业出版社,2000.
[2] Bovet,Cesati. 深入理解 Linux 内核[M]. 陈莉君,等. 北京:中国电力出版社,2001.

作者简介:

高珍(1978-),女,研究生,计算机软件与理论专业;吴永明(1946-),男,教授,研究生导师;周卫华(1976-),女,研究生,研究方向为计算机网络。