第34卷 第22期 计 算 机 工 程 2008年11月
Vol.34 No.22 Computer Engineering November 2008
·网络与通信· 文章编号：1000—3428(2008)22—0097—03 文献标识码：A 中图分类号：TP393

基于预约的证书撤销通知方案

黄 河，王亚弟，韩继红
(解放军信息工程大学电子技术学院，郑州 450004)

摘 要：分析Ad Hoc网络中证书撤销通知方案的优缺点，提出一种基于预约的证书撤销通知方案。当节点的证书状态发生变化时能主动及时地把证书的最新状态通知给预约该证书状态的所有节点，通过单向哈希链实现预约及证书状态通知消息的认证，同时利用自恢复区域方法传播预约及证书状态通知消息，并运用Jini技术实现该方案。实验结果表明，该方案是有效的。

关键词：Ad Hoc网络；证书撤销；预约；自恢复区域；哈希链

Certificate Revocation Notification Scheme Based on Subscription

HUANG He, WANG Ya-di, HAN Ji-hong
(School of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004)

【Abstract】The advantages and drawbacks of some existed certificate revocation notification schemes in Ad Hoc network are analyzed, and a certificate revocation notification scheme based on subscription is proposed. When the node's certificate state changes, the latest state of the certificate will be actively informed to all nodes which subscribe this certificate state in time. The identification of subscription and certificate state notification messages is implemented by using one-way Hash chain, and these messages are diffused with self-healing community method. Moreover, this scheme is realized by Jini technique. Experimental results show this scheme is effective.

【Key words】Ad Hoc network; certificate revocation; subscription; self-healing community; Hash chain

1 概述

Ad Hoc网络是种特殊的移动无线通信网络，它不依赖现有的网络基础设施，通过节点间的相互协作迅速组网。网络中所有节点地位平等，无需设置任何控制节点，具有很强的抗毁性。动态变化的网络拓扑、无线通信、分布式控制等特征使Ad Hoc网络相比传统有线网络更易遭到攻击，如窃听，身份伪造等。

节点身份的认证是安全通信的前提，数字证书机制是实现节点身份认证的一个有效途径，但随着时间的推移，节点证书或是被更新，或是被撤销，而通信期间节点又必须确保通信对方的数字证书始终有效(否则，当节点的数字证书被撤销时，与该节点的通信就会被终止)。因此，如何及时得知节点证书的最新状态对于确保Ad Hoc网络的安全应用具有实际意义。

2 相关工作

在有线网络中，节点可通过查询CA分发的CRL列表或利用在线证书状态查询协议访问在线数据库，从而得到节点证书的最新状态。由于Ad Hoc网络无法部署集中式CA，也不可能设置一个节点来提供证书状态查询服务(否则，这样的节点一旦失效，整个Ad Hoc网络将无法正常工作，破坏其抗毁性)，因此这种机制不适用于Ad Hoc网络。另外，当利用OCSP协议访问节点证书时，节点需要不停地查询以确保通信对方的证书始终有效，而当多个节点同时与一个节点通信时，数据库节点就需要处理多个请求，这些请求其实是针对同一节点证书状态的请求，这样的冗余操作会影响协议运行的效率，不适用于节点资源有限的Ad Hoc网络。

现有的Ad Hoc网络安全方案[1-3]大多没有就节点证书状态变化问题进行分析研究，其他方案根据提供节点证书状态信息的节点个数分为2类：单节点型和全节点型。这2类方案都存在一些问题：单节点型以文献[4]中的方案为代表，设定一个能周期性访问的CA，利用从CA收到的CRL列表查询得到节点的证书状态，该方案存在单节点失效问题；全节点型[5-6]即每个节点都维护一张包含网络中所有节点证书状态信息的信息表，并通过不断更新该信息表得到节点的最新状态。其中，文献[5]和文献[6]的区别在于撤销证书的方式不同，前一种方案利用合法节点的控诉次数，后者基于节点的权值，当权值为0时，节点的证书被撤销。文献[7]提出一种基于哈希链的证书撤销方案，节点可以根据证书的相关信息计算其有效性，但该方案使用被动等待的证书撤销方式，无法及时撤销非法节点的证书。单节点型的证书状态获取方案维护开销小，但安全性不足；全节点型的证书状态获取方案具有较强的容侵性，但通信及维护开销较大，实用性不强。本文综合以上两者的优点，提出一种多节型的证书状态获取方案，该方案可以主动把节点证书的最新状态通知给预约该证书的节点。

3 基于预约的证书撤销通知服务

基于预约的证书撤销方案利用分布式认证技术，由服务节点(拥有系统私钥份额，本文称为仓库节点)维护节点的证书信息，确保信息的可信性及可用性。使用自恢复区域方法传递预约消息及证书状态通知消息[8]，确保消息成功传递。

作者简介：黄 河(1982－)，男，硕士研究生，主研方向：计算机网络安全；王亚弟，教授、博士生导师；韩继红，副教授

收稿日期：2008-02-25 **E-mail：**amos412328@yahoo.com.cn

另外，由于Ad Hoc网络中节点资源有限，因此利用单向哈希链[9]代替数字签名实现消息源认证可以节省节点的计算开销，增强Ad Hoc网络的可持续性。

3.1 基于自恢复区域方法的消息传递

自恢复区域方法的本质是利用节点冗余实现数据包的成功传递。自恢复区域示意如图 1 所示。

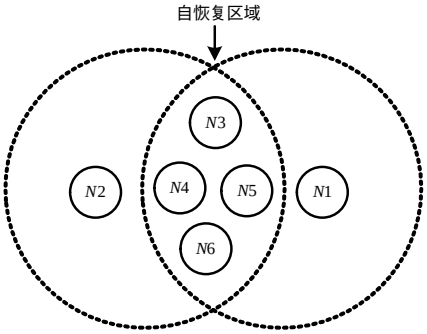

图 1 自恢复区域

节点N_1和N_2通信范围交集内的所有节点构成自恢复区域。由于自恢复区域中的每个节点既能与节点N_1直接通信又能与节点N_2直接通信，并且能监视区域中其他邻居节点的行为，因此自恢复区域中只要有一个节点是合法的就可以将N_1要发送的数据转发给N_2。

随着通信距离的增加，利用自恢复区域方法传递消息的效率会越来越低，但本方案的应用前提是基于分簇的 Ad Hoc 网络，且单个簇中节点间的最大距离是 2 跳，因此，该方法对本方案有较强的实用性。

3.2 基于单向哈希链的消息源认证

消息源认证通常由数字签名实现，但计算量大。由于 Ad Hoc 网络中节点计算资源有限，因此基于数字签名的消息源认证不适合 Ad Hoc 网络。单向哈希链是种拥有类似于公钥技术性质的密码技术，且计算效率高。本方案采用单向哈希链实现 Ad Hoc 网络中的消息源认证。

用户先选择一个随机数 r 作为种子，然后用一个单向哈希函数对随机数 r 作 m 次递归运算，得到序列：

$$r, h(r), h^2(r), \cdots, h^l(r), \cdots, h^{m-1}(r), h^m(r)$$

其中，r 类似于公钥技术中的私钥；$h^m(r)$ 称为哈希链的根节点，类似于公钥技术中的公钥，知道 $h^m(r)$ 而不知道 r 就无法计算出 $h^l(r)$，而给定 $h^l(r)$，对它作 $m-l$ 次递归哈希运算，同时与 $h^m(r)$ 比较就能验证其正确性及来源。初始时节点对 $h^m(r)$ 进行签名，并发送到其他节点。

使用单向哈希链进行消息源认证的消息有 4 种：

(1)预约消息。节点 i 向仓库节点 j 预约节点 k 证书状态的消息为 $subscrb(i,j,k)$，其格式为

$$subscrb(i,j,k) = (PK_i, PK_j, PK_k, sub, index, h^{m-index}(r), date, ttl)$$

其中，PK_i, PK_j, PK_k 分别为节点 i、节点 j 和节点 k 的公钥；sub 是对预约消息的描述；$date$ 为发出该消息的时间；ttl 为该预约消息的有效期。收到该消息的节点只需对 $h^{m-index}(r)$ 作 $index$ 次递归运算并与 $h^m(r)$ 比较即可实现消息源认证。为了认证的安全性及有效性，$index$ 随着预约消息数量的增加呈单调递增。

(2)预约反馈消息。当仓库节点 j 发现节点 k 的证书失效时，若节点 i 对节点 k 证书状态的预约还未过期，则向节点 i 发送预约反馈消息以告知节点证书状态的变化，该消息为

$$acksub(i,j,k) = (PK_i, PK_j, PK_k, flag, index, h^{m-index}(r), date)$$

其中，$flag$ 标识节点证书的状态。1 表示证书有效，0 表示证书被撤销。

(3)请求确认消息。当节点 i 收到某仓库节点 j 关于节点 k 证书被撤销的预约反馈消息时向其他仓库节点发送关于证书 k 的请求确认消息，该消息为

$$affirm(i,k) = (PK_i, PK_k, affirm, index, h(r)^{m-index}, date)$$

其中，$affirm$ 为请求确认信息。

(4)确认反馈消息。当仓库节点 s 收到请求确认消息后检查状态信息表，然后向节点 i 发送确认反馈消息：

$$ackaff(s,i,k) = (PK_s, PK_i, PK_k, flag, ack, index, h^{m-index}(r), date)$$

其中，ack 为确认反馈消息标示符。

3.3 基于预约的证书撤销通知方案

图 2 为仓库节点提供预约服务的流程。

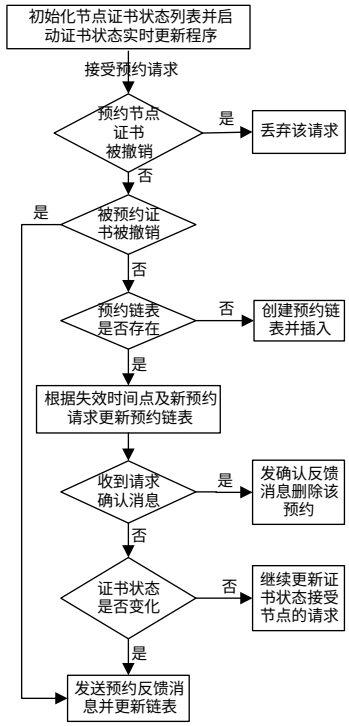

图 2 仓库节点提供预约服务流程

仓库先初始化证书列表，根据预约消息建立预约链表。当收到请求确认消息或证书状态发生变化时发送反馈消息并更新预约链表，证书状态的更新贯穿整个流程。

基于预约的证书撤销通知方案包括 5 个阶段：初始化阶段，接受预约请求阶段，请求链表维护阶段，反馈阶段和确认阶段。

(1)初始化阶段。通过仓库节点完成证书状态列表的初始化，列表由节点身份 ID 号，节点公钥及证书状态组成，证书状态随着时间的推移不断更新，证书的更新由某种分布式证书撤销算法实现。

(2)接受预约请求阶段。当一个节点 i 与另一节点 k 通信时，系统向周围服务节点预约节点 k 的证书，当仓库节点 j 收到预约请求时先检查节点 i 和节点 k 的证书状态，若节点 i 的证书已被撤销，则忽略此请求；若节点 k 的证书已被撤销，则直接向 i 发送反馈消息；若 2 节点的证书均有效，则检查节点 k 的证书预约链表是否存在，若不存在，则创建预约链

表并写入节点 i 的请求，包括节点 ID 号及失效时间点；若存在，则按失效时间的先后顺序把节点 i 的预约请求插入到预约链表中。

(3)请求链表维护阶段。在节点证书变化前，根据预约请求失效时间点不断删除已失效的请求，并按失效时间的先后顺序插入新的请求。

(4)反馈阶段。当节点 k 的证书被撤销时，系统向节点 k 的证书状态预约链表中的所有节点发送反馈消息，以告知节点 k 的证书已被撤销，同时释放节点 k 的预约链表。

(5)确认阶段。若节点 i 收到某仓库节点 j 关于预约节点 k 证书的反馈消息，即节点 k 的证书被撤销，则节点 i 暂停与节点 k 的通信，并向其他仓库节点发送请求确认消息。如果收到的消息中多数仓库节点已撤销节点 k 的证书，则终止与节点 k 的通信，否则恢复通信，并发起对仓库节点 j 的控诉消息。

4 方案实现技术

Jini 技术通过引入租约的概念来解决资源或服务的释放问题。被租用的资源或服务的授权是基于时间的，一旦租借期满，服务就将结束，资源被释放。租约的期限在第 1 次授权时规定由租约的授权者和接收者采用 request/response 方式协商。租约机制是通过接口类 Lease 提供的，其 UML 图形如图 3 所示。

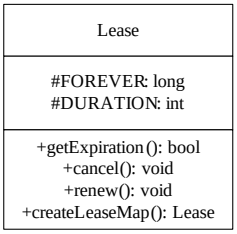

图 3 接口 Lease 的 UML 示例

其中，getExpiration 方法用来判断租约是否到期；cancel 方法用来取消租约；renew 方法用来续订租约；类型 LeaseMap 用来解决多个用户请求同一资源或服务的问题。

普通节点通过 Jini 技术的 lookup 机制查找撤销通知服务，然后添加 remoteEventListener(LeaseEvent)产生基于时间的证书预约，仓库节点接到远程事件请求后，初始化 Java Space 对象空间，以接受节点进程的注册，当预约的证书被撤销时，该空间就产生一个事件并发送给已登记的进程。

5 结束语

本文提出一种基于预约的证书撤销通知方案，由多个仓库节点维护 Ad Hoc 网络中所有节点的证书状态，节点能够向仓库节点预约其他节点的证书。当被预约证书的状态发生变化时，该节点会被主动告知，这种方案融合了单节点型和全节点型证书状态获取方案的优点，可以有效避免多个节点对同一证书状态的重复性查询。另外，节点可以通过向多个仓库节点预约某节点证书状态，以确保及时得到该证书的最新状态，并能避免得到错误的状态信息。此方案效率较高，具有很强的实用性。

参考文献

[1] Seung Y, Kravets R. MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks[C]//Proc. of the 2nd Annual PKI Research Workshop Program. Gaithersburg, Maryland, USA: [s. n.], 2003.

[2] Mohamed E, Lamia B, Farouk K. A Totally Distributed Cluster Based Key Management Model for Ad Hoc Networks[C]//Proc. of the 3rd Annual Mediterranean Ad Hoc Networking Workshop. Bodrum, Turkey: IEEE Computer Society, 2004.

[3] Dong Y. Provding Distributed Certificate Authority Service in Cluster Based Mobile Ad Hoc Networks[J]. Computer Communications, 2007, 30(11): 2442-2452.

[4] Morogan M, Muftic S. Certificate Management in Ad Hoc Networks[C]//Proc. of Symposium on Applications and the Internet Workshops. Orlando, Florida, USA: IEEE Computer Society, 2003.

[5] Zerfos P. URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc NetWorks[J]. IEEE/ACM Trans. on Networking, 2004, 12(6): 1049-1063.

[6] 宁红宙, 刘 云, 何德全. 一种用于 Ad Hoc 网络的分布式证书撤销算法[J]. 北京交通大学学报, 2005, 29(2): 44-47.

[7] 赵志新, 张浩军, 杨 峰, 等. 一种适用于 Ad Hoc 网络的高效证书撤销机制[J]. 计算机应用与软件, 2006, 23(10): 128-130.

[8] Hong Xiaoyan, Yi Yunjun. A Secure Ad Hoc Routing Approach Using Localized Self-healing Communities[C]//Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing. Washington, USA: ACM Press, 2005.

[9] 赵源超, 李道本. 一种新颖的可再生多哈希链的构造[J]. 电子与信息学报, 2006, 28(2): 299-302.

(上接第 88 页)

5 结束语

本文针对应用软件二进制程序给出了转换到可读性强的中间语言的程序转换方法，主要转换技术包括汇编文法设计、代码分析、汇编语言到中间语言映射以及代码优化。该方法可供软件分析人员在维护和操作软件时作为高级调试工具使用，由此可以极大地减少工作人员在代码安全分析问题上追踪代码所需的工作量。文中描述方法可以在没有程序源码和任何调试信息的情况下对程序进行转换，而且在此基础上可以更加方便地进行各种复杂数据类型的恢复和控制结构分析，最终实现向高级语言转换的目标，由此该方法具有更大的实用性。该方法的实现达到了较高的自动化程度，转换后代码量明显减少，代码表示更接近于高级语言，可读性强，易于理解，效果理想。

参考文献

[1] Program Transformation Wiki. Program Transformation[EB/OL]. (2005-09-10). http://www.program- transformation.org.

[2] 陈凯明, 刘宗田, 任传胜. 逆编译中面向用户的中间语言设计与实现[J]. 小型微型计算机系统, 2002, 23(10): 1173-1176.

[3] 陈凯明, 刘宗田. 反编译研究现状及其进展[J]. 计算机科学, 2001, 28(5): 113-115.

[4] Allen F E, Cocke J. A Program Data Flow Analysis Procedure[J]. Communications of the ACM, 1976, 19(3): 137-147.

[5] Cifuentes C. Reverse Compilation Techniques[D]. Queensland, Australia: Queensland University of Technology, 1994.

[6] Data Rescue. IDA Pro Disassembler[EB/OL]. (2006-10-20). http://www.datarescue.com/idabase.