对2-3树证书撤销方法的改进*)

黄迎春[1] 何良生[2] 蒋 凡[1]

(中国科技大学计算机科学技术系 合肥230027)[1] (信息安全国家重点实验室 北京100039)[2]

摘 要 2-3树的证书撤销方法在通信和计算开销上有很多优点,随着节点的增加,该方法也有一些不足,如调整树的计算开销就很高。用证书的生效时间计算出证书的剩余有效期,此对撤销证书集进行划分,每个子集建立一棵2-3树。我们称之为剩余有效期游标树,它降低了原来的树高。此外,对于撤销证书集里过期的证书,认证机构和目录不必进行删除操作,减小了通信和计算开销。

关键词 证书,剩余有效期游标树,CRL,2-3树

An Improved Method of Certificate Revocation Based on 2-3 Tree

HUANG Ying-Chun HE Liang-Sheng JIANG Fan

(Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230027)[1]

(The State Key Laboratory of Information Security, Beijing 100039)[2]

Abstract There are many advantages of 2-3 tree method for certificate revocation in communication costs and computation costs. However, with the tree nods increasing, the method also shows some shortages, such as the costs of tree adjust will be too high. Based on the beginning time of the certificate to be valid the certificate's remainder valid period can be counted easily and then be used to partition the set of the certificates. We build a 2-3 tree in each subset. It is called cursor tree of remainder valid priod and much shorter than the primary tree. Moreover, for the overdue certificate in the revocation set, CA and directory need not delete it, so the communication and computation costs can also be reduced.

Keywords Certificate, Cursor tree of remainder valid period, CRL, 2-3tree

1 概述

在开放环境中,保证实体间传递的信息保密、可靠、完整,是网络应用追求的目标之一。非对称密码体制的提出和完善为这一目标的工程化实现奠定了基础。公钥基础设施(PKI-Public key Infrastucture)即此工程化实现的框架。PKI 用证书确保公钥的真实性,证书是由公共信任机构(CA-Certification Authority)签名发布的信息,它包括公钥和一些附加的数据,如序列号、有效期等。

当一个证书发布后,有多种情况可能使一个证书在它的有效期满前被撤销,包括名字的改变、主体和 CA 间的关联改变、私钥受到威胁等。现行的证书撤销方法多采用 PKIX[2]工作组提供的证书撤销列表(CRL-Certificate Revocation List)方法,这种方法的主要优点是简单;缺点是发送整个 CRL 的通信开销大,且用户没有简捷的验证其证书有效性的方法。

基于 CRL 提出许多改进的方法,如 Delt-CRL[2,3]、分段 CRL[2,3]等,但它们都没有根本解决 CRL 的缺点;Kocher 提出了基于哈希树[4]的证书撤销树(CRT-Certificate Revocation Tree)[5]方法。Naor 和 Nissim 在 CRT 方法的基础上又提出了2-3树证书撤销方法[1]。它有很好的安全性和可扩展性,在通信和计算上有很多优点,但要删除撤销证书中过期的部分。我们提出了一种改进的2-3树证书撤销方法——剩余有效期游标树证书撤销方法,它简单地完成撤销证书中过期部分的删除而无需任何开销。

2 2-3树证书撤销方法

CRT 是 VeliCert 公司的 Paul Kocher 开发的一项技术,CRT 方法中证书撤销树的叶子表示未撤销的证书序号的区间(如图1),它用撤销证书的序列号(设有 n 个撤销证书,将它们按序排列),这 n 个撤销证书把所有证书分成了 $n+1$个区间,用这 $n+1$个区间作为树的叶子节点,构造一棵哈希树。如撤销证书的序列号分别为35、48、67、85,那么它们划分的区域为:$n_{0,0}(-\infty,34)$、$n_{0,1}(35,47)$、$n_{0,2}(48,66)$、$n_{0,3}(67,84)$、$n_{0,4}(85,\infty)$,如图1所示,最左一列为叶子,右边为其上一级节点,同一级的相邻节点合二为一用哈希函数 $h()$计算出上一级节点,如果没有成对的节点,剩余的节点直接进入下一级,直到算出最后的根节点为止。判断一个证书是否被撤销,只需检查该证书的序列号是否是叶子节点区间表达式的左端即可。

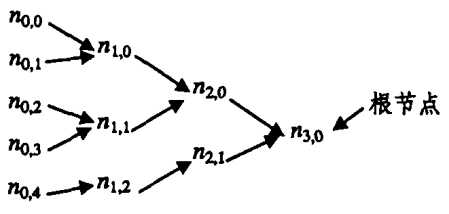

图1 CRT 示例图

对于树的更新(插入、删除)会使叶子分裂和合并,因为 CRT 是个二叉树,如果更新发生在最小的叶子上,则会引起

*)本文受到国家"863"计划项目(2003AA148050)资助。黄迎春 硕士研究生,研究方向为信息安全。何良生 研究员,研究方向为信息安全。蒋凡 教授,研究方向为协议工程、信息安全。

对所有叶子的重新划分，从而要重新计算整个树。

基于认证撤销树的结构，Naor 和 Nissim 提出了一种改进的方法——2-3树证书撤销方法。它用2-3树作为认证撤销树结构，树的叶子对应撤销证书的序列号(有序的)，每个中间节点有2到3个孩子(它的值由其孩子的哈希函数给出)，证明一个证书是否撤销变成证明树中是否存在一个特定的叶子：

·如果证书撤销，则证明存在相应的叶子。

·如果证书 X 未撤销，则证明存在相邻的叶子 X_1、X_2($X_1<X<X_2$)。

3 对2-3树方法的改进

2-3树法对过期的撤销证书需要进行删除操作，在系统达到稳定时，新增加的撤销证书数量和过期的撤销证书数量是相同的，所以删除的开销是更新开销的一半。在保留2-3树法安全、可扩展等特性的基础上，我们的方法消除了删除的开销，又因为压缩了树的高度，所以还减小了插入的计算开销和通信开销(在本文的讨论中，计算开销是指为完成树的调整应用 h()进行计算的开销；通信开销是指树调整后 CA 到目录传送的验证数据的通信开销和目录响应用户查询的通信开销)。

基于 Naor 和 Nissim 在文[1]中对2-3树方法的描述，本节介绍剩余有效期游标树方法。

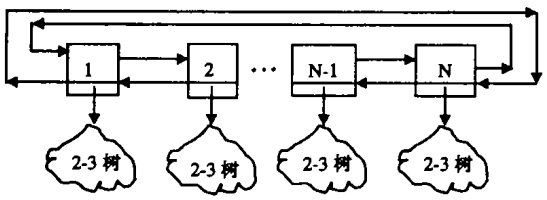

图2 剩余有效期游标树的结构

3.1 对树结构的改进

如图2，为讨论方便，设证书的有效期为一年(考虑闰年的情况，取 N＝366)，先建立一个双向链表，表中共有366元素，每个元素称作一个剩余有效期游标。游标下为一棵2-3树，树中撤销证书的剩余有效期是相同的。游标的计算如下：每过一天，各个游标的值减1，值为1的游标变为0，表明在它下面的撤销证书已经过期，删除并把它们送入归档设备中，并将链表中值为0的游标变成值为366的游标，产生一个新的游标。对于证书的查询和验证，计算出它的剩余有效期 R，到值为 R 的游标下的子树中验证即可。下面给出此改进方法的描述。

3.2 子树的生成和修改

对树的每个节点赋值：

·用撤销证书的序列号(有序的)作为树的叶子节点。

·对于中间节点，应用冲突难解哈希函数 h()(在我们的实验中用文[6]提供的 MD4程序实现此函数)作用于它的孩子，算出该节点的值。

修改过程：

·删除：对已过期的撤销证书，CA 和目录每天将游标值为0的子树中的撤销证书移入归档设备中。

·插入：在相应的子树中插入新的撤销证书的序列号，并修改插入路径上的节点的值。

3.3 CA、目录和用户的操作

我们讨论的问题是撤销证书的发布、查询、及有效性的验证，主要涉及 CA、目录和用户，参照文[7]对它们进行说明：CA 是可信的第三方，对证书和撤消证书签名。目录是用于存储和发布公钥证书和证书撤销列表。客户是证实数字签名和验证来自认证机构的已知公钥的证书路径的有效性的实体。

CA 的操作：

·创建证书：CA 通过对包含证书数据、证书序列号和有效期的消息签名来产生一个证书。

·初始化：如图2所示，先建立一个含366个元素的双向链表，每个元素(游标)分别对应1，2…，365，366。然后，将最初被撤销的证书，按其生效的时间计算出它的剩余有效期，并放到相应游标下的子树中，如：证书的生效时间为2003-10-5，现在的时间为2003-12-5，那么，此证书的剩余有效期为304，将它放到游标304下面的2-3树中。然后，CA 计算和保存所有游标下的树的节点并把撤销的证书序列号列表(有序的)随同包含对游标标记、各子树的树根值、树高的列表和时间戳的签名送到目录中。

·修改：每过一天，将各个游标的值减1，将游标值为0的子树中的撤销证书删除并送入归档设备中，同时将游标值变为366。对于插入操作，先计算出证书的剩余有效期，再插到相应的游标下的子树中。为了修改目录要重新计算被修改的树，CA 将新增的撤销证书列表和带有证书所在不同游标下的新的树根值、树高的列表和时间戳的签名送到目录上，如：列表中证书的剩余有效期分别为：10、31、31、280、306、330、330，那么，在新增撤销证书的列表中加入10、31、280、306、330游标标记，并且将游标值为10、31、280、306、330的游标下的树根值、树高和时间戳签名后送到目录上。

目录的操作：

·初始化：目录首先建立一个含366个游标的双向链表，通过接受 CA 初始的撤销证书列表，自己计算构建整个剩余有效期游标树，检验树根值、树高列表中对应的游标下的子树和时间戳；并核查 CA 对于这些值的签名。

·对 CA 修改的响应：每过一天，将各个游标的值减1，将游标值为0的游标下的树摘除，同时将游标值变为366。按照从 CA 接收到的增加列表，根据游标标记对剩余有效期游标树做修改，计算所有受影响的节点值，检验时间戳并按树根值、树高列表中的值检验相应的子树。

·对用户查询的请求(只查询相应游标下的2-3树)：为回答用户的查询，目录给用户发送带有树根值、树高时间戳的签名。1)如果被查询的证书是已撤销的，目录给用户发送从根到叶子(被查询的证书)的路径上的节点和它们孩子的节点值。2)如果被查询的证书(序号为 L)未被撤销，给用户发送到两个相邻叶子的路径，设相邻的叶子为 L_1，L_2($L_1<L_2$)，那么，$L_1<L<L_2$。

用户的操作：

用户先检验 CA 对证书的签名并检查证书的有效期。然后向目录送证书的序列号 L。收到目录回答后，检验 CA 在树根值、树高和时间戳上的签名。1)如果目录指明查询的证书已被撤销，那么用户用哈希函数 h()检查目录提供的从叶子到根的路径。2)如果目录指明查询的证书未被销，用户检查目录提供的两条路径并检查它们所通向的两个相邻的叶子($L_1<L_2$)，检查 $L_1<L<L_2$是否成立。

4 分析

下面先说明分析所用到的参数：

·n—证书的总数。

·k—每个 CA 平均拥有的证书数。

·p—证书在过期前被撤销的概率($p=0.1$)(设证书的有效期为一年366天,则每天撤销的证书数为$\frac{n\cdot p}{366}$)。

·q—每天需要查询状态的证书数。

·T—每天修改的次数。

·λ_{sn}—证书序列号占有的 bit 数($\lambda_{sn}=128$)。

·λ_{sig}—签名的长度($\lambda_{sig}=1,000$)。

·λ_{hash}—哈希函数所需的安全参数($\lambda_{hash}=128$)。

·C_{hash}—进行一次哈希的计算量。

·λ_{root}—树根值的 bit 数($\lambda_{root}=128$)。

·$\lambda_{height\cdot cursor}$—树高值和游标标记的 bit 数($\lambda_{height\cdot cursor}=5+9=14$bit,树高小于32,游标的数目<512)。

C_{hash}、λ_{root}、$\lambda_{height\cdot cursor}$为新增的参量,其它为文[1]中参量。

表1 2-3树法和剩余有效期游标树法的通信开销和计算开销的比较

		2-3树法	剩余有效期游标树法
通信开销	CA 到目录	$2\cdot\frac{n\cdot p\cdot\lambda_{sn}}{366}+T\cdot\lambda_{sig}$	$\frac{n}{k}[\frac{k\cdot p\cdot\lambda_{sn}}{366}+\lambda\cdot(\lambda_{height\cdot cursor}+\lambda_{root})]+T\cdot\lambda_{sig}$
	目录到用户	$2\cdot q\cdot\lambda_{hash}\cdot\log_2(p\cdot k)$	$2\cdot q\cdot\lambda_{hash}\cdot\log_2(\frac{p\cdot k}{366})$
计算开销	CA	$2\cdot\frac{n\cdot p}{366}\cdot C_{hash}\cdot\log_2 k$	$\frac{n\cdot p}{366}\cdot C_{hash}\cdot\log_2\frac{k}{366}$
	目录	$2\cdot\frac{n\cdot p}{366}\cdot C_{hash}\cdot\log_2 k$	$\frac{n\cdot p}{366}\cdot C_{hash}\cdot\log_2\frac{k}{366}$

下面对表1进行一些说明。2-3树方法中 CA 到目录的通信开销:每天证书在过期前被撤销的个数为$\frac{n\cdot p}{366}$,这些证书要增加到目录中;撤销证书中每天过期的个数为$\frac{n\cdot p}{366}$,这些证书要从目录中删掉。

改进方法中 CA 到目录的通信开销:λ 为不同游标的数目,$1\leqslant\lambda\leqslant366$,在最坏的情况下,即各游标下的树都有新增的撤销证书,要送366个树的高度和树根值,但它是一个常量,当$k>1,470,000$时,$365\cdot\frac{n}{k}\cdot(\lambda_{height\cdot cursor}+\lambda_{root})<\frac{n\cdot p\cdot\lambda_{sn}}{366}$,即改进方法传送的 bit 数要小于2-3树的 bit 数。

5 实验

用5台 PC 机(配置为 PVI1.6G/60G/512,模拟一个目录,三个 CA,一个客户)建一个实验环境,它们的操作系统是 REDHAT 9.0,为了统计传送数据,用自己实现的 SOCKET 程序完成它们之间的通信,以512字节为单位。取证书撤销的概率为0.1(估计每年证书的撤销数量为10%[8]),目录上有500,000节点,三个 CA 中,两个各有200,000节点,一个有100,000节点,树的叶子节点用随机生成的128bit 的十六进制数表示如:0105 C7E2 36CF 1853 063C DCD7 53BE 5DAC,树的中间节点用文[6]提供的 MD4程序生成的128bit 数(时间戳和数字签名对两种方法是相同的,所以试验中未使用它们)。对于有两个值的中间节点,它的第二孩子要参加两次 MD4运算,它的两个值在向其父节点做 MD4运算时连接为一个256bit 的值。我们用了两周内的平均数据。

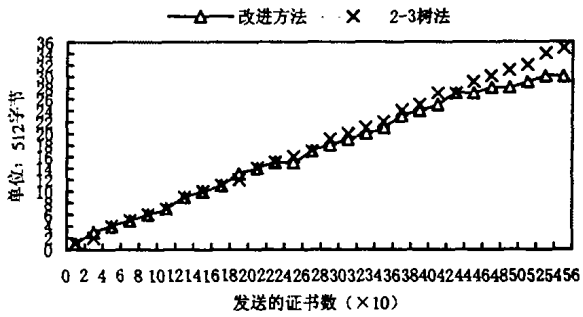

图3 撤销证书为200,000的 CA 到目录的通信开销比较

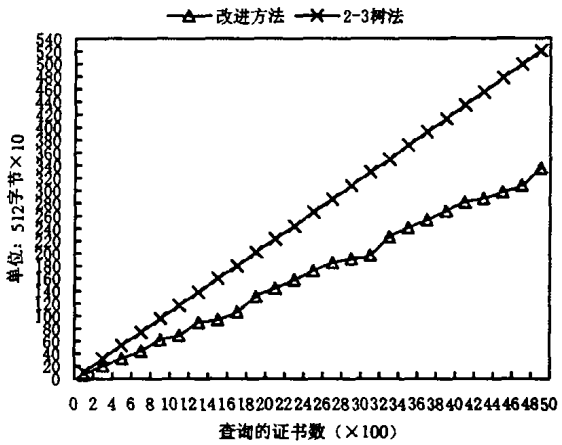

图4 撤销证书为500,000的目录到客户的通信开销比较

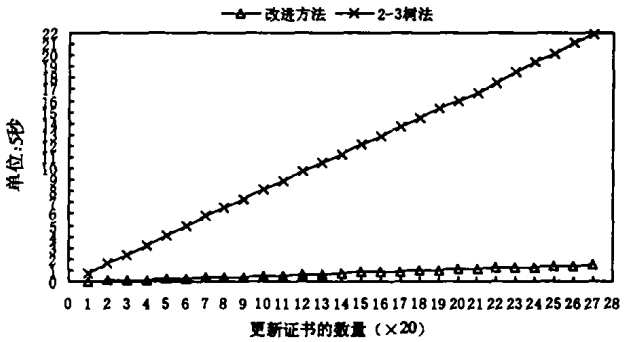

图5 撤销证书为200,000的 CA 中的计算开销比较(目录中的计算开销同此)

注:1)2-3树方法的更新撤销证书的平均数量是改进方法的两倍,所以上面各图中用2-3树法中的两个证书对应改进方法的一个证书。2)两种方法中,对于过期撤销证书的处理有一点区别,在2-3树法中,每次传送更新中,都要搜索过期的撤销证书,并把对它们的删除加入到更新列表中,在我们改进的方法中,每天只在一个时间(我们取23:59,要求 CA、目录和用户的系统时间应该正确)把当天的过期的撤销证书一次性移走。

结论 当 CA 的节点数大于1,470,000时,我们的方法在 CA 到目录的通信开销上,优于2-3树方法。当 CA 的节点数小于1,470,000时,我们的方法和2-3树法基本相同(见图3)。

在目录到用户的通信开销上,我们的方法比2-3树法节省了$2\cdot q\cdot\lambda_{hash}\cdot\log_2 366$。

在 CA、目录的计算开销上,我们的方法比2-3树法节省了$\frac{n\cdot p}{366}\cdot C_{hash}\cdot\log_2 366\cdot k$。

证书的生效时间是每个证书的一个很重要的特征量,用

它划分一个大的集合，可以节省删除的开销，还可以减小树结构撤销证书方法发布和验证的通信开销和计算开销。

参考文献

1 Naor M, Nissim K. Certificate Revocation and Certificate Update. In: Proc. of the 7th USENIX Security Symposium San Antonio, Texas, Jan. 1998. 26～29. http://www.usenix.org/publications/library/proceedings/sec98/full_papers/nissim/nissim.pdf

2 Housley R, Ford W, Polk W. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. April 2002. http://www.ietf.org/rfc/rfc3280.txt

3 Berkovits S, Herzog J C. A Comparison of Certificate Validation Methods for Use in a Web Environment: [MITRE Ttechnical Report]. Nov. 1998. 12～13 http://citeseer.nj.nec.com/460528.html

4 Merkle R C. A Certified Digital Signature. In: Proc. Crypto '89, Lecture Notes in Computer Science 435, Springer Verlag, 1989. 234～246

5 Kocher P C. On Certificate Revocation and Validation. FC'98, LNCS 1465, 1998. 172～177

6 Rivest R. The MD4 Message Digest Algorithm. MIT Lab for computer science Internet RFC 1186 Oct. 1990 http://www.faqs.org/rfcs/rfc1186.html

7 NIST PKI Project Team. Minimum Interoperability Specification for PKI Components, Version2 - Second DRAFT, Aug. 31, 2000. http://csrc.nist.gov/pki/documents/MISPC2_public3_20000831.pdf

8 Arens A. Public Key Certificate Revocation Schemes. Feb. 2000. 14 http://citeseer.nj.nec.com/arnes00public.html

（上接第46页）

表1 测试例的动态部分

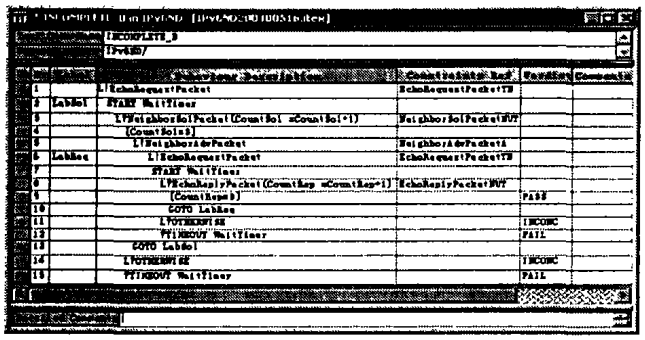

表2 邻居宣告消息定义

Solicited flag	Override flag	LLA equal to cache	Neighbor Advertisement
set	set	no	A
set	set	yes	B
set	clear	no	C
set	clear	yes	D
clear	set	no	E
clear	set	yes	F
clear	clear	no	G
clear	clear	yes	H

·网络侦包工具：Sniffer4.60.01

·被测试系统：Windows2000 + SP3 + tpipv6-001205-SP3-IE6.zip

·测试拓扑图：如图3所示

下面是我们对 Windows2000 + SP3上 IPv6实现的邻居发现协议的节点间状态转换关系的一致性测试报告，限于篇幅，我们只对在测试过程中发现的一些不一致性给出说明，如表3所示。

表3 邻居发现协议不一致性测试报告

Test Case Name	NA	Old State	New State (RFC)	New State (实测)	Update LLA (RFC)	Update LLA (实测)
NC_ReachC	C	REACHABLE	STALE	REACHABLE	NO	NO
NC_DelayC	C	DELAY	DELAY	REACHABLE	NO	YES
NC_DelayG	G	DELAY	DELAY	DELAY	NO	YES
NC_ProbeC	C	PROBE	PROBE	REACHABLE	NO	YES
NC_ProbeG	G	PROBE	PROBE	PROBE	NO	YES

结束语 文中给出了一种验证IPv6邻居发现协议一致性的形式化方法，即用 FSM 和 MSC 相互结合的方法为协议建模，然后用测试例描述语言(TTCN)对测试例进行具体的描述。有了用 TTCN 描述的测试例以后，我们就可以在自己的协议测试执行系统上对测试例进行解释执行，对协议实现进行一致性测试。实践中，我们用自己的协议测试执行系统，对 Windows + SP3上 IPv6邻居发现协议实现的一致性进行了测试。目前，基于本文提出的建模方法生成测试例的过程中，需要手工干预的内容较多，进一步的工作是提高此测试例生成模型的自动化生成程度以及进行优化以减少测试例的数目。

参考文献

1 ISO/IEC 9646-1. Open Systems Interconnection-Conformance testing methodology and framework - Part 1: General Concepts, 1994

2 ISO/IEC 9646-2. Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite Specification, July 1988

3 ISO/IEC 9646-3. Open Systems Interconnection-Conformance testing methodology and framework - Part 3: The Tree and Tabular Combined Notation (TTCN), 1994

4 ISO/IEC 9646-4. Open Systems Interconnection - Conformance testing methodology and framework - Part 4: Test Realization, June, 1988

5 Deering S, Hinden R. RFC 2460 Internet Protocol, Version 6 (IPv6) Specification. S. Deering, R. Hinden. Dec. 1998

6 Narten T, Nordmark E, Simpson W. RFC 2461 Neighbor Discovery for IP Version 6(IPv6). Dec. 1998

7 叶新铭，王红霞，石立新. 下一代网络协议一致性测试执行系统的实现. 计算机科学，2003，30(4)：51～54

8 田军，张玉军，余东，等. 邻居发现协议的形式化测试. 计算机研究与发展，2001，38(12)

9 毕军，史美林. 计算机网络协议测试及其发展. 电信科学，1996，12：51～54