证书吊销的线索二叉排序 Hash 树解决方案*

王尚平[1,2]，张亚玲[2]，王育民[1]

[1](西安电子科技大学 ISN 国家重点实验室，陕西 西安 710071)；

[2](西安理工大学 计算中心，陕西 西安 710048)

E-mail: spwang@mail.xaut.edu.cn

http://www.xaut.edu.cn

摘要：提出了公钥基础设施(public key infrastructure，简称 PKI)中证书吊销问题的一个新的解决方案——线索二叉排序 Hash 树(certificate revocation threaded binary sorted hash tree，简称 CRTBSHT)解决方案. 目前关于证书吊销问题的主要解决方案有 X.509 证书系统的证书吊销列表(certificate revocation list，简称 CRL)、Micali 的证书吊销系统(certificate revocation system，简称 CRS)、Kocher 的证书吊销树(certificate revocation tree，简称 CRT)及 Naor-Nissm 的 2-3 证书吊销树(2-3CRT)，这些方案均不完善. 在 CRT 系统思想的基础上，利用线索化二叉排序树及 Hash 树给出的新方案，既继承了 CRT 证明一个证书的状态(是否被吊销)不需要整个线索二叉树，而只与其中部分相关路径有关的优点，又克服了 CRT 在更新时几乎需要对整个树重新构造的缺点，新方案在更新时仅需计算相关部分路径的数值. 新方案对工程实现具有一定的参考价值.

关键词：公钥基础设施；证书权威；公钥证书；证书吊销；数字签名

中图法分类号：TP309 **文献标识码**：A

电子商务安全支付中的一个重要课题是对各个实体身份的认证，而对实体身份的认证是通过公钥基础设施(public key infrastructure，简称 PKI)技术来实现的，即将某一实体与其所持的公钥通过公钥证书捆绑在一起. 这样对实体身份的认证，实际上是对其所持有的公钥证书的认证. 公钥证书是由公钥基础设施中具有公共信任资格的证书权威(certification authority，简称 CA)签名的一组信息. 该组信息除了含有该公钥持有人的身份信息及公钥参数等数据以外，还有该证书编号及证书有效期限等. 公钥基础设施通过公钥证书链来传递信任，进而实现对各个实体身份进行认证的目的.

当证书颁发之后，其有效性通常由有效期加以限制. 但经常由于密钥泄露，也可能仅仅是怀疑密钥泄漏或证书持有人的工作岗位变动等原因，有些证书需在其失效期前吊销. 信誉卡系统是这方面的一个类似的典型例子，当一个卡被盗或丢失时，卡的持有人报告信誉卡管理中心，管理中心必须在该卡失效前予以吊销. 持有证书只是证明该证书有效的必要条件而非充分条件. 因此，需要建立一种证书吊销及证书有效性查询的机制，以维护系统的安全运行.

证书吊销中主要涉及证书权威、名录服务(directory)和证书用户(user). CA 是该系统中普遍信任的一方，具有一个系统用户皆知的公钥证书，负责颁发和担保其所签名证书的可信任性. 用户首先注册，在用户身份通过审核后，由 CA 签发公钥证书捆绑用户身份和其持有的公钥，并通过证

* 收稿日期：2000-12-25；修改日期：2001-05-08

基金项目：国家自然科学基金资助项目(60073052)；陕西省教育厅自然科学研究计划资助项目(00JK266)

作者简介：王尚平(1963－)，男，陕西扶风人，博士生，副教授，主要研究领域为信息保密理论，电子商务的安全性；张亚玲(1966－)，女，陕西西安人，讲师，主要研究领域为计算机软件开发；王育民(1936－)，男，北京人，教授，博士生导师，主要研究领域为信息论，信道编码，密码学，通信网的安全.

书吊销机制维护系统的正常运转.CA 在线可能受到攻击,且 CA 可能成为通信瓶颈,因此 CA 一般不直接对用户提供在线证书查询,CA 通过周期性更新名录服务,由名录服务负责用户证书有效性的查询.名录服务是一个数据库,存放 CA 发布的有关证书吊销的信息,可使用户高效访问,为用户提供证书有效性查询信息.用户经常需查询与业务有关的其他用户的证书的有效性(这时查询用户为商家的角色),或得到所持有证书有效性的一个证明,并将此证明附在其证书后面作为有效性的证明,该证书的接受者不再需要和名录服务联系,即可验证该证书的有效性.用户访问名录服务查询证书是否被吊销,名录服务负责回答用户的查询,但名录服务是不被信任的一方,故名录服务回答查询并要给出该回答的有效性的证明,即证明该回答与 CA 存放在名录服务中的信息一致,且用户可验证该证明的真实性,故证明中要包含被信任方 CA 的签名信息.一旦用户的秘密泄露或怀疑秘密泄露或其他情况,用户可以及时通知 CA 吊销其所持有的证书,由 CA 通过证书吊销机制,通知其他用户该证书已被吊销.

目前,关于证书吊销问题的解决方案主要有 X.509 证书系统的证书吊销列表(certificate revocation list,简称 CRL)、Micali 的证书吊销系统(certificate revocation system,简称 CRS)、Kocher 的证书吊销树(certificate revocation tree,简称 CRT)以及 Naor-Nissim 的 2-3 证书吊销树(2-3CRT).

CRL 原理简单,当商家查询一个证书时,名录服务需将最新的全部 CRL 传递给商家,这样导致了名录服务到商家的查询通信成本过高.CRS 则采用周期性地给出全部证书状态的证明方案,这样导致 CA 到名录服务的通信成本过高.CRT 则是利用 Hash 树思想构造证书状态证明.CRT 与 CRL 相比的主要优点是证明一个证书的状态不需要全部 CRT,而只需其中相关路径的部分.且证书拥有者可拥有其证书有效性的简明证明.但其主要缺点是对 CRT 的更新计算量过大,被吊销证书的任何一个插入或删除都可能会导致整个 CRT 的重新计算.

本文在 CRT 系统思想的基础上,基于线索化二叉排序树及 Hash 树方法,提出了证书吊销问题的一个新的解决方案——线索二叉排序 Hash 树解决方案 CRTBSHT(certificate revocation threaded binary sorted hash tree).新方案继承了 CRT 的优点,即证明了一个证书的状态不需要整个线索二叉树,而只需要其中相关路径的部分,同时在更新该树时,不需要对整个树重新计算,仅需计算相关路径,为此所付出的代价是树中结点记录比 CRT 复杂,包含线索化信息.

本文第 1 节是 CRL,CRS 和 CRT 的相关知识.第 2 节是我们提出的证书吊销问题的线索二叉 Hash 树解决方案.第 3 节是对新方案与 CRT 的分析和讨论.第 4 节是结束语.

1 CRL,CRS 和 CRT 简介

CRL 是 X.509 证书系统[1]中关于证书吊销问题的现行标准解决方案.CRL 包含所有被吊销证书的序列号(serial number)的列表,并含有发行日期、失效日期、CA 的数字签名等信息.CRL 由 CA 发行更新,更新的 CRL 由 CA 周期性地发送给名录服务,名录服务在回答一个查询时,将最新的 CRL 全部传送给查询者,若某个证书的序列号在 CRL 中,则查询者确认该证书已被吊销,否则在证书的有效期内该证书有效.CRL 的主要优点是方案简单,但主要问题是名录服务对商家的通信成本过高,因为 CRL 列表可能变得非常庞大,每次查询时需将 CRL 列表全部传给查询者,且证书拥有者不能得到其拥有证书有效性的一个简明的直接证明.

为了改善名录服务到查询者的通信代价,Micali 提出了证书吊销系统 CRS[2].其主要思想是 CA 周期性地利用单向哈希函数对全部证书状态(即是否有效)给出一个证明,但对全部证书的周

期性证明造成了 CA 到名录服务的通信量的大大增加,且 CA 到名录服务的通信成本与更新率成正比,故 CRS 方案限制了吊销证书的更新率(Micali 取更新周期 $T=1$ 天). 本文的思想与 CRS 不同,我们的思想更接近于 CRT. 下面介绍一下 CRT 方案.

Kocher 提出了 CRT 方案[3],主要目的是证书验证者能得到一个证书状态(有效还是吊销)的一个简短证明. CRT 是 Hash 树,其叶子为一些声明. 例如,设 CA 吊销了两个证书,其证书号分别为 X_1 及 X_2,且 $X_1<X_2$,则其声明为"若 CA_x-CA 且 $X_1\leqslant X<X_2$,则 X 被吊销,当且仅当 $X=X_1$". CRT 由这样一些声明为叶子的二元哈希树生成. 一个证书状态的证明是由其相应的叶子到根结点的路径上所有结点及其子节点的 Hash 值构成. 下面用文献[3]中的例子简单说明 CRT. 设 CA 发行的证书中编号为 5,12,13,15,20,50,99 的证书被吊销,该 CRT 的构造如图 1 所示.

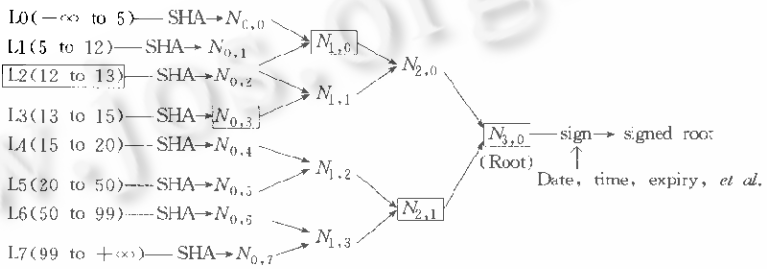

Fig. 1 Sample certificate revocation tree
图1 CRT构造示例

假设要证明编号为 12 的证书被吊销,则需提供从叶子 L2(12 to 13)到根结点路径上的相关数据,图中以加框表示.

CRT 的主要缺点是更新成本过高. 更新是指最新被吊销的证书号需要加入 CRT 中,或原来 CRL 中被吊销证书因到达其失效期而自动失效者需从 CRT 中删除. 假如有一个证书新被吊销,则 CRT 树几乎全部需要重新计算,因为其采用了证书号的完全顺序排列.

Naor-Nissm 在文献[4]中给出了 CRT 的变异方案 2-3CRT,将 CRT 由二叉树改为 2-3 树,以解决 CRT 更新时整个树需要重新计算的问题. 但 2-3 树的结构会给实际操作带来困难,其数据结构不如二叉树简单,对有效证书的证明也不够简明.

为此,本文引入线索化二叉排序树的解决方案,新方案继承了 CRT 的优点,即证明了一个证书的状态不需要整个线索二叉树,而只需要其中相关路径的部分(大约为所有结点数的对数个数值). 同时,在更新该树时,不需要对整个树重新计算,仅需计算相关路径,且保留二叉树的结构,对有效证书可给出简明证明,较好地解决了证书吊销问题.

2 证书吊销问题的线索化二叉排序 Hash 树解决方案

首先给出证书吊销问题的抽象化模型. 这里,我们采用文献[4]中的定义.

定义. 设 U 是一个全集,S 是 U 的一个子集,$S\subset U$,设 D_S 是代表 S 的一个数据结构.

• 设 $\langle e\rangle$ 代表对 $e\in S$ 的成员查询,其回答为 $\langle a\rangle$,$a\in\{\text{YES},\text{NO}\}$,分别对应于 $e\in S$ 或 $e\notin S$. 若其回答为 $\langle a,p\rangle$,其中 a 同上,且 p 是由 CA 签名的一个关于 a 的证明,则称这样的查询为成员认证查询.

• 更新运算是指如下两种运算:

(1) $\langle Insert,e\rangle$,其中 $e\in U\backslash S$,则插入 e 后的数据结构为 $D_{S'}$,这里 $S'=S\cup\{e\}$;

(2) $\langle Remove,e\rangle$,其中 $e\in S$,则删除后的数据结构为 $D_{S'}$,这里 $S'=S\backslash\{e\}$.

下面设 U 是 CA 发行的全部证书编号集,S 是 CA 吊销的证书编号集,我们用二叉排序树及其线索信息以及 Hash 树来构造数据结构 D_S,并讨论其认证查询以及插入、删除运算.

2.1 证书吊销二叉排序 Hash 树

首先设 CA 吊销的证书的编号集为 $S=\{n_0,n_1,\ldots,n_{m-1}\}$,为了使二叉树保持基本的平衡性,这里设 n_0 是一个被吊销证书号的中间值,其余的证书编号按吊销的顺序任意排列.下面以证书编号为关键字,以 n_0 为根结点,对 S 构造排序二叉树[5],使每个结点的数据结构为

LC	Num	Parent	RC

其中 LC,RC 分别是指向其左右子树的根结点的指针,Parent 为指向其父结点的指针.

上述结点结构可以采用如下的类型定义:

```
TYPE bitreep=↑bnode;
     bnode=RECORD
                Num:datatype;
                LC,RC,Parent:bitreep
           END
```

这样,按二叉排序树建立的标准算法[5]可以建立证书吊销二叉排序树 p.该树的每个结点对应于一个吊销证书.利用强抗碰撞 Hash 函数(MD5[6]或 SHA-1[7]等)计算每个结点的 Hash 值,每个结点的 Hash 值为其左右子树根结点的 Hash 值及该结点的证书号的 Hash 运算的结果值.最后,CA 对根结点的 Hash 值在加入时戳等信息后进行签名,得到吊销证书二叉排序 Hash 树(CRBSHT).这样得到的树反映了被吊销的证书的信息,类似于 CRT,证明某个证书被吊销只需要给出该证书对应的结点到根结点的路径及相关结点的 Hash 值即可.查询者在收到证明以后,可利用 Hash 函数计算得到根结点的 Hash 值,并利用 CA 的公钥与证明中 CA 对根结点的签名验证名录服务应答证明的有效性.但吊销证书二叉排序 Hash 树仅反映了被吊销证书的信息,对有效证书不能给出其有效性的简明证明.为了完善吊销证书二叉排序 Hash 树,我们进一步对其进行了中序线索化,构造了线索化二叉排序 Hash 树,使其含有未被吊销证书(有效证书)区间列表,这样,对有效证书也可给出一个简明的证明.

2.2 证书吊销线索化二叉排序 Hash 树(CRTBSHT)

为了证明一个证书的有效性,考虑将证书吊销二叉排序树 p 进行线索化操作.为此构造证书吊销的线索化二叉排序树结点结构如下:

$L[n_1,n_2]$	LC	Ltag	Num	Parent	Rtag	RC	$R[n_3,n_4]$

其中,Num 为该结点对应的废止证书编号,Parent 为指向其父结点的指针,LC 和 RC 分别是指针类型,Ltag 为左标志,Rtag 为右标志,$L[n_1,n_2]$和 $R[n_3,n_4]$为有效证书的区间(可能为空(∅)),其定义如下:

$Ltag=0$,表示该结点的左子树不空且 Lc 指向的是该结点的左子树的根,并令 $L[n_1,n_2]$为空(∅).

$Ltag=1$,表示该结点的左子树为空,令 Lc 指向的是该结点的中序前驱结点,并令 $L[n_1,n_2]$为有效证书的区间,其中 n_1 为 Lc 指向的结点的 Num 域加 1,n_2 为当前结点的 Num 域减 1.若 $n_1>n_2$,令 $L[n_1,n_2]$为空(∅),此时当前结点的证书号(Num 域)为前驱结点的证书号(Num 域)加 1(两者相连).

$Rtag=0$,表示该结点的右子树不空且 Rc 指向该结点的右子树的根,并令 $R[n_3,n_4]$为空(∅).

$Rtag=1$,表示该结点的右子树为空,令 Rc 指向该结点的中序后继结点,且 $R[n_3,n_4]$为有效证

书的区间,其中 n_3 为当前结点的Num域加1,n_4 为Rc指向的该结点的中序后继结点的Num域减1.若 $n_3>n_4$,令 $R[n_3,n_4]$ 为空(∅),此时当前结点的证书号(Num域)为后继结点的证书号(Num域)减1(两者相连).

这样,对二叉排序树中左子树为空的结点,利用LC域记录其中序前驱,并增加 $L[n_1,n_2]$ 记录有效证书编号的区间,而对二叉排序树中右子树为空的结点,利用RC域记录其中序后继,并增加 $R[n_3,n_4]$ 记录有效证书编号的区间.其中Ltag和Rtag反映了线索化信息.当 $Ltag=0$ 时,Lc指向其左子树的根;当 $Ltag=1$ 时,Lc指向该结点的前驱结点.当 $Rtag=0$ 时,Rc指向其右子树的根;当 $Rtag=1$ 时,Rc指向该结点的中序后继结点.

线索化的吊销证书编号二叉排序树的结点为如下结构:

```
TYPE thread=↑bnode;
  bnode=RECORD
          Num:datatype;
          L[n1,n2],R[n1,n2]:stringtype;
          Lc,Rc,Parent:thread;
          Ltag,Rtag:booltype;
        END
```

下面给出由二叉排序树构造线索化二叉排序树的算法,在算法中同时完成了有效证书区间的计算(设 p 指向证书吊销二叉排序树的根).

算法1.

```
Proc Inthread(p)
Begin
  If p≠nil then
    [ Inthread(p↑.Lc); {左子树线索化}
      If p↑.Lc=nil then
        [p↑.Ltag:=1;p↑.Lc:=pre;
         p↑.L[n1,n2]:=[pre↑.num+1,p↑.num-1]
        ]
      If pre↑.Rc=nil then
        [pre↑.Rtag:=1;pre↑.Rc:=p;
         pre↑.R[n3,n4]:=[pre↑.num+1,p↑.num-1]]
       pre:=p;  {保持pre指向p的前驱}
      Inthread(p↑.Rc)
    ]
END
```

调用算法(完成对吊销证书中最小证书号的前驱和最大证书号的后继的特殊处理).

```
new(pre); pre↑.num:=-1;
Inthread(p);      {p为指针,指向要线索化的二叉排序树的根}
pre↑.Rtag:=1;pre↑.Rc:=nil;
 pre↑.R[n3,n4]:=[pre↑.num+1,+∞);
```

以第1节的例子作为吊销证书编号集 $S=\{50,12,5,99,20,13,15\}$,以50为根结点,以吊销证书编号作为关键字,形成二叉排序树,如图2所示.

利用算法1及其调用算法可以得到图2的线索化二叉排序树,如图3所示.其中用虚线箭头表示结点的中序线索化信息.为了直观,当结点的 $Ltag=1$ 时,有效证书的区间 $L[n_1,n_2]$ 表示为一个左叶子,用虚线连接.同理,当 $Rtag=1$ 时,有效证书的区间 $R[n_3,n_4]$ 表示为一个右叶子,用虚线连接.

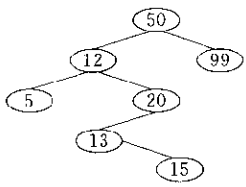

Fig. 2 Sample certificate revocation binary sorted hash tree
图2 证书吊销二叉排序树示例

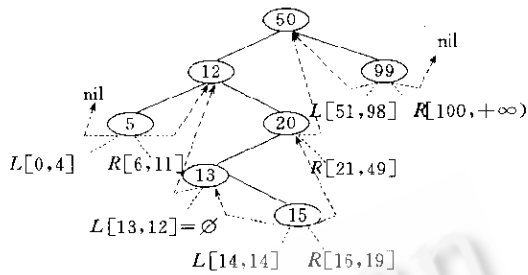

Fig. 3 Sample certificate revocation threaded binary sorted hash tree
图3 证书吊销线索化二叉排序树

在图3中,当 $n_1>n_2$ 时,叶子区间 $L[n_1,n_2]$ 表示为空($\varnothing$),当 $n_1=n_2$ 时,$L[n_1,n_2]$ 仅为一个点.

2.3 证书吊销线索化二叉排序树的 Hash 树

为了保证上述线索化证书吊销二叉树的数据结构 D_S 信息的完整性,我们利用抗碰撞 Hash 函数(MD5[6]或 SHA-1[7]等)计算每个结点的 Hash 值,每个结点的 Hash 值为其左子树根结点 Hash 值(若存在左子树)或该结点的有效证书的区间 $L[n_1,n_2]$(若不存在左子树)级联该结点的证书号再级联该结点右子树根结点 Hash 值(若存在右子树)或该结点的有效证书的区间 $R[n_3,n_4]$(若不存在右子树)的 Hash 运算的结果值. 最后,CA 对根结点的 Hash 值在加入时戳等信息后进行签名,得到吊销证书线索化二叉排序 Hash 树(certificate revocation threaded binary sorted hash tree,简称 CRTBSHT). 这样得到的树反映了全部证书的信息,包括被吊销证书及有效证书. 证明某个证书被吊销只需给出该证书对应的结点到根结点的路径和相关结点的 Hash 值以及 CA 对根结点的签名即可,而证明某个证书有效只需给出该证书所在的有效证书区间所在的结点到根结点的路径和相关结点的 Hash 值以及 CA 对根结点的签名即可. 查询者在收到证明后可利用 Hash 函数计算得到根结点的 Hash 值,并利用 CA 的公钥验证证明中 CA 对根结点签名的有效性.

对线索化证书吊销二叉树的数据结构 D_S 进行上述 Hash 计算,并存储 Hash 值,即得到证书吊销线索化二叉排序树的 Hash 树. 这样,结点数据结构中要增加一个 Hash 域,增加后的结点数据结构为

$L[n_1,n_2]$	Lc	Ltag	Num	Parent	Rtag	Rc	$R[n_3,n_4]$	Hash

Hash 域的值为一个 bit 字符串(Hash 函数为 MD5时为128bit,Hash 函数为 SHA-1时为160bit),Hash 域计算的公式表示为 $p\uparrow.Hash:=hash(H(PL)\|p\uparrow.num\|H(PR))$,其中

$$H(PL)=\begin{cases}hash(P\uparrow.LC.\uparrow Hash), & \text{当 } p\uparrow.Ltag=0\text{(左子树存在)时}\\ hash(P\uparrow.L[n_1,n_2]), & \text{当 } p\uparrow.Ltag=1\text{(左子树不存在)时}\end{cases},$$

$$H(PR)=\begin{cases}hash(P\uparrow.RC.\uparrow Hash), & \text{当 } p\uparrow.Rtag=0\text{(右子树存在)时}\\ hash(P\uparrow.R[n_1,n_2]), & \text{当 } p\uparrow.Rtag=1\text{(右子树不存在)时}\end{cases},$$

且运算符号‖表示级联运算.

这样,CA 只需对根结点的 Hash 值进行签名,即可保证对整个数据结构完整性的认证. 任何对证书吊销线索化二叉排序 Hash 树的改动都可被检测出来,除非可以找到 Hash 函数的一个碰撞. 当然,数字签名[6]时要加入时戳、新鲜性以及有效期间,以防止重放攻击.

一个成员认证查询〈e〉是查询 $e\in S$,名录服务接到该查询后,按二叉排序树的搜索规则搜索 e,若搜索成功,则 e 是一个被吊销的证书编号,名录服务回答 YES,并给出证明 p. p 是由证书编号

(Num域)为e的结点到线索化二叉排序Hash树的根结点的路径上所有点及其左右子树根结点的数据值以及CA对根的Hash值签名构成.若e是一个有效的证书编号,则$e\notin S$,但e应属于某个结点的有效证书区间$L[n_1,n_2]$或$R[n_3,n_4]$,此时名录服务回答NO,并且给出证明p.p为该结点到线索化二叉排序Hash树的根的路径上所有结点及其每个左右子树根结点的数据值以及CA对根的Hash值的签名.用户收到该证明后,对其根结点的Hash值进行计算,并利用CA的公钥与证明中CA对根的Hash值签名进行验证,并检验e确实属于一个有效证书区间,若验证一致,则接受证明,否则名录服务的回答不可信.名录服务伪造证明相当于名录服务找出Hash函数的一个碰撞,由于采用的是强抗碰撞Hash函数,这几乎是不可能的(伪造成功的概率可以忽略).

插入运算$\langle Insert,e\rangle$根据二叉排序树的特点进行.若e在插入前为一个有效证书编号,则必位于属于某个结点的有效证书区间$L[n_1,n_2]$(或$R[n_3,n_4]$)之内,首先查找到该结点,e作为该结点的左(或右)子树的根结点插入,并修改该结点的标志、线索化信息及有效证书区间,并需重新计算从插入结点到根结点的路径上的相关Hash数据.

删除运算$\langle Remove,e\rangle$,首先执行查找运算.设查找成功时p指针指向待删结点,则需分4种情况处理.(1)若该结点为叶结点(即$p\uparrow.Ltag=1$且$p\uparrow.Rtag=1$),则删除该结点并仅需修改其父结点$p\uparrow.Parent$的标志、线索化信息及有效证书区间,并需重新计算从叶结点到根结点的路径上的相关数据.(2)若该结点只有左子树,即$p\uparrow.Ltag=0$且$p\uparrow.Rtag=1$,这时如果$p\uparrow.Num<p\uparrow.Parent\uparrow.Num$,则令$p\uparrow.Parent\uparrow.Lc:=p\uparrow.Lc$,如果$p\uparrow.Num>p\uparrow.Parent\uparrow.Num$,则令$p\uparrow.Parent\uparrow.Rc:=p\uparrow.Lc$,删除该结点并修改$p\uparrow.Parent$及$p\uparrow.Lc$的相关数据,并需重新计算从$p\uparrow.Lc$结点到根结点的路径上的相关Hash数据.(3)若该结点只有右子树,即$p\uparrow.Ltag=1$且$p\uparrow.Rtag=0$,这时如果$p\uparrow.Num<p\uparrow.Parent\uparrow.Num$,则令$p\uparrow.Parent\uparrow.Lc:=p\uparrow.Rc$;如果$p\uparrow.Num>p\uparrow.Parent\uparrow.Num$,则令$p\uparrow.Parent\uparrow.Rc:=p\uparrow.Rc$,删除该结点并修改$p\uparrow.Parent$及$p\uparrow.Rc$的相关$Hash$数据,并需重新计算从$p\uparrow.Rc$结点到根结点的路径上的相关Hash数据.(4)若该p结点的左右了树均非空,即$p\uparrow.Ltag=0$且$p\uparrow.Rtag=0$,则查找p结点的左子树的最右下结点f,用f结点取代p结点,并将$f\uparrow.Lc$赋给$f\uparrow.Parent.Rc$,同时修改线索化信息及从$f\uparrow.Parent$结点到根结点的路径上的相关Hash数据.

3 讨 论

成功利用二叉排序树的结点插入及删除运算仅影响原来的树的一个分枝,保持原树的其他分枝不变,是本方案的特色,也是对CRT的最大改进.图1和图3是对同一证书吊销问题采用CRT方案和本文所提的CRTBSHT方案的图示,显然,图3包含了全部证书的状态信息.CRT方案中一个插入或删除会导致全部树的重新计算,本文提出的CRTBSHT方案仅需计算相关路径上的数据.且本文提出的CRTBSHT结构的结点数比CRT少一半,这可由本方案的线索二叉排序Hash树的结点数仅与CRT的叶子结点数相等这一点看出.本文提出的新方案CRTBSHT与2-3CRT方案相比,新方案为二叉树,其结构简单,2-3CRT树的结构相对复杂.2-3CRT的缺点之一是当查询一个证书状态时,回答为一个有效证书时的证明是回答为一个被吊销证书的数据量的2倍[4],而实际上绝大多数证书是有效证书,而新方案则不存在这样的问题.在新方案中,CRTBSHT树对有效证书及被吊销证书的证明回答数据量相当.2-3CRT树中从叶子到根结点的路径长度是固定的,这是2-3CRT树的优点,新方案中的路径长度则不统一,有的结点到根结点路径很短,有的则可能很长,而2-3CRT则不出现这一问题.路径可能变得很长的问题可通过路径长度策略加以解决,即对路径

长度规定一个界,一旦路径长度超过这个界,则调整根结点重新建立全部树一次,维持树的平衡性,保持路径不超过规定的界.

4 结束语

随着公钥密码体制越来越广泛的使用,CA 发行证书的吊销问题将成为公钥基础设施(PKI)中日益重要的课题.构造快速、高效的证书吊销机制具有重要意义.当电子商务中证书系统广泛使用时,证书吊销机制将是维护系统安全的重要保证.本文提出的 CRTBSHT 方案既继承了 CRT 的优点,又克服了 CRT 的缺点,是一个值得考虑的证书吊销方案.

References:

[1] Hously, R., Ford, W., Polk, W., *et al*. Internet X.509 public key infrastructure certificate and CRL profile. IETF RFC 2459, 1999. http://www.ietf.org/rfc/rfc2459.html.

[2] Micali, S. Efficient certificate revocation. Technical Memory, MIT/LCS/TM-5426, 1996. http://www.lcs.mit.edu/publications.

[3] Kocher, P. On certificate revocation and validation. In: Hirschfeld, R., ed. Financial Cryptography-FC'98. LNCS 1465, Berlin: Springer-Verlag, 1998. 171~177.

[4] Moni, Naor, Kobbi, Nissim. Certificate revocation and certificate update. IEEE Journal on Selected Areas in Communications, 2000,18(1):561~170.

[5] Yan, Wei-min, Wu, Wei-min. Data Structure. Beijing: Tsinghua University Press, 1991. 118~133 (in Chinese).

[6] Rivest, R.L. The MD-5 message digest algorithm. RFC1321, Internet Activities Board, 1992. http://www.ietf.org/rfc/rfc1321.html.

[7] FIPS 180-1. Secure Hash Standard. Washington, DC: Department of Commerce, National Institute of Standard and Technology, 1995. http://csrc.nist.gov/cryptval/shs.html.

附中文参考文献:

[5] 严蔚敏,吴伟民.数据结构.北京:清华大学出版社,1991.118~133.

Threaded Binary Sorted Hash Trees Solution Scheme for Certificate Revocation Problem*

WANG Shang-ping[1,2], ZHANG Ya-ling[2], WANG Yu-min[1]

[1](*National Key Laboratory on ISN, Xidian University, Xi'an* 710071, *China*);
[2](*Computing Center, Xi'an University of Technology, Xi'an* 710048, *China*)
E-mail: spwang@mail.xaut.edu.cn
http://www.xaut.edu.cn

Abstract: A new solution scheme called certificate revocation threaded binary sorted Hash trees (CRTBSHT) for certificate revocation problem in public key infrastructure (PKI) is proposed in this paper. Previous solution schemes including traditional X.509 certificates system's certificate revocation lists (CRL), Micali's Certificate Revocation System (CRS), Kocher's Certificate Revocation Trees (CRT), and Naor-Nossim's 2-3 certificate revocation trees (2-3CRT), but no one is perfect. The new scheme keeps the good properties of CRT that it is easy to check or prove whether a certificate is revoked which only needs the related path values but does not need the whole CRT values and overcomes the disadvantage of CRT that any update will cause the whole CRT to be computed completely. The new scheme has referential value to the PKI engineering practice.

Key words: public key infrastructure; CA (certification anthority); public key certificate; certificate revocation; digital signature

* Received December 25, 2000; accepted May 9, 2001

Supported by the National Natural Science Foundation of China under Grant No. 60073052; the Natural Scientific Research Plans of Educational Department of Shanxi Province of China under Grant No. 00JK266