第 35 卷 第 5 期 计 算 机 工 程 2009 年 3 月
Vol.35 No.5 Computer Engineering March 2009
·安全技术· 文章编号：1000—3428(2009)05—0168—03 文献标识码：A 中图分类号：TP301.6

基于 Dijkstra 的 PKI 交叉认证路径搜索算法

熊 熙，高 飞
(北京理工大学信息科学技术学院电子工程系，北京 100081)

摘 要：针对网状型公钥基础设施(PKI)信任模型认证路径的不确定性，提出一种基于 Dijkstra 算法的 PKI 交叉认证路径搜索算法。该算法根据 PKI 系统中配置的认证路径搜索服务器，结合信任路径图，利用 Dijkstra 算法进行认证路径搜索。结果表明，该算法在一定程度上避免了认证路径的不确定性，有效提高了路径查找的速度。

关键词：公钥基础设施；Dijkstra 算法；交叉认证；认证路径

Path Searching Algorithm for PKI Cross-certification Based on Dijkstra

XIONG Xi, GAO Fei
(Department of Electronic Engineering, School of Information Science and Technology, Beijing Institute of Technology, Beijing 100081)

【Abstract】Aiming at the uncertainty of certification path of the network-like Public Key Infrastructure(PKI) trust model, this paper brings forward a path searching algorithm for cross-certification. According to path searching servers distributed in PKI system and confidence-path maps, the algorithm searches the shortest certification path with Dijkstra algorithm. Analysis shows that, to some extent, this solution avoids the uncertainty of the certification path, and successfully accelerates the searching.

【Key words】Public Key Infrastructure(PKI); Dijkstra algorithm; cross-certification; certification path

2002 年~2003 年，IETF 工作组起草制定了一系列的公钥基础设施(Public Key Infrastructure, PKI)标准规范，如 RFC3280, RFC3379。目前一些草案还在不断修整，如 X.509v4 新增了很多支持交叉认证的扩展项。RFC3280 虽然已经详细定义了证书链的通用验证算法，但是并没有提供一个统一的证书路径构造算法和搜索算法，因此，认证路径的搜索还需要进一步研究[1]。在中型 PKI 系统中，域间常用网状信任模型进行交叉认证，如果每次都直接颁发证书，效率必然很低。本文改进了网状 PKI 信任模型，并将 Dijkstra 算法应用于 PKI 认证模型中，提出了一种 PKI 认证路径搜索算法。该算法充分考虑了认证路径的长度问题，尽量利用已建立的交叉认证提高整个 PKI 系统交叉验证的效率，最后分析了该算法的复杂度。

1 PKI 的信任模型和 Dijkstra 算法

PKI 信任模型通常可分为树型、网状型和桥型。树型 PKI 信任模型的证书路径容易扩展，但对根证书机构(Certification Authority, CA)的安全性要求很高。网状型 PKI 信任模型具有很好的灵活性，但存在从用户证书到可信任点建立证书的路径不确定问题。桥型 PKI 可以克服 2 种基本 PKI 结构的缺点，并可连接不同结构的 PKI 系统，但由于它包含了部分树型和网状型 PKI，因此也存在上述 2 种 PKI 模型的缺点[2]。

Dijkstra 算法的主要思想是按照增加的路径成本逐步确认离源节点最近的节点，这是一种迭代算法。在第 1 次迭代中，算法找到离源节点最近的节点，如果链路成本是正数，该节点必定是源节点的相邻节点。在第 2 次迭代中，算法查找离源节点次近的节点，该节点必定是源节点的相邻节点，或者是离源节点最近节点的相邻节点；否则，将存在一个更近的节点。依此类推，在第 k 次迭代中，算法会确定第 k 个离源节点最近的节点[3]。

本文将 Dijkstra 算法应用于 PKI 认证模型中：各域的根 CA 相当于 Dijkstra 算法中带权图的节点，查找 2 个根 CA 间认证的最短路径相当于查找带权图中两节点间的最短路径。

2 网状模型的改进及其算法设计

2.1 网状模型的改进

本文的设计方案适用于中型规模的网络认证。在中型规模的网络认证中，如果采用单一层次结构，即在各个原来的根 *CA* 之上再设置一个根 *CA* 来为哪些根 *CA* 颁发证书，则新的根 *CA* 数据处理压力过大，而且找一个原根 *CA* 都信任的再上一级 *CA* 并不一定成功。因此，在维持原来各个域的信任关系的基础上，各个域的根 *CA* 之间采用交叉认证。本文针对这种情况对现有的网状模型做了改进，如图 1、图 2 所示。

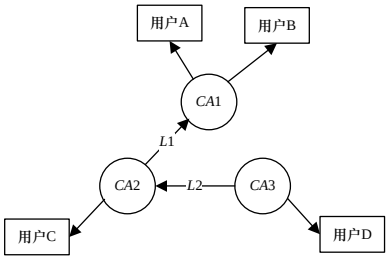

图 1 *CA*3 通过 *CA*2 对 *CA*1 认证的模型

基金项目：国家部委基金资助项目
作者简介：熊 熙(1983－)，硕士，主研方向：信息安全；高 飞，教授
收稿日期：2008-08-29 **E-mail：**flybear1983@sina.com

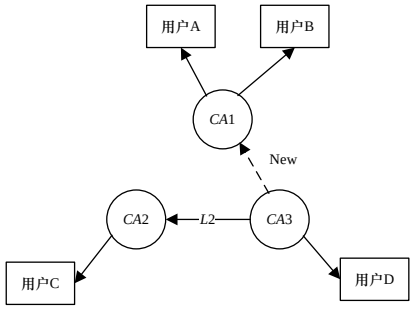

图 2　*CA*3 向 *CA*1 直接颁发证书的模型

在改进的网状模型中，由于认证过程中必须加入成本，因此不采用广度优先算法和深度优先算法；由于无需计算所有根 *CA* 对之间的最短路径，因此采用加权图中的一种最短路径查找算法，即 Dijkstra 最短路径算法。

对于改进的网状模型，在单一域中建立层次认证的信任路径并不复杂，但对于域间信任路径的建立，由于网状结构使其证书路径搜索问题成为搜索一对根 *CA* 间的路径问题，因此在实现上采用以下技术：

(1)通过 Dijkstra 算法查找已有路径。不一定要在 2 个信任域间直接建立交叉认证，而是先动态地查找已有路径[4]，如图 1 所示，如果 *CA*3 要认证 *CA*1，则可以通过 *CA*2 去认证 *CA*1。如果不存在路径，则在必要时再构建交叉认证，如图 2 所示。

(2)在每个根 *CA* 建立一个存放域间根节点交叉认证关系的目录服务器，并在一个 PKI 系统中配置一个或多个证书路径搜索服务器[5]，其实际地址在各根 *CA* 中声明。在路径搜索服务器中保存一个由证书路径信息节点构成的域节点间信任路径图，以便在图中实现优化查找(域间)路径[4]。目录服务器中要存放根间的交叉认证证书和本域内该根节点以下的所有证书。

(3)各个域的根 *CA* 之间的认证关系随时间变化，在每次发生证书撤销或有效期延长等情况时，其余各个 PKI 系统都需要得到通知，以改变其路由数据(证书的验证成本及传递成本)。

2.2　算法描述

假设 2 个节点(根 *CA*)间的传递成本与认证方向无关，在有向加权图中，设 *Cij* 表示从 *i* 节点到 *j* 节点的最小成本，*Bi* 表示证书在节点 *i* 认证的成本，*Dij* 为节点 *i* 到节点 *j* 的当前最小成本，则

$$Dij = Bj+Cij$$

设 *Di* 为源节点到节点 *i* 的当前最小成本，*N* 表示当前节点集合，该集合由那些最短路径已确定的、被永久标记的节点组成，各个根 *CA* 在向临近根 *CA* 签发交叉认证证书时，已经获得相邻根 *CA* 的传递成本和验证成本。本文的证书路径搜索算法步骤如下：

(1)初始化：

$$N=\{s\}$$

其中，*s* 为源节点。

$$Dj=Csj + Bj$$

其中，所有 $j \neq s$。

$$Ds=Bs$$

其中，*Bs* 为验证 *B* 的自签名证书。

(2)查找下一个最近的节点。寻找节点 *i* 不属于 *N*，使得

$$Di = \min Dj$$

如果存在这样的 *i*，将其加入到 *N* 中。如果 *N* 包含目标节点，则算法结束。

(3)在节点 *i* 加入到 *N* 后，更新最小成本。对于每个节点 *j*(*j* 不属于 *N*)：

$$Dj = \min \{Dj, Di+Cij+Bj\}$$

(4)转步骤(2)。

上述认证算法的核心是通过 Dijkstra 算法[3]在已有的认证关系中查找最佳路径。网状认证模型中的交叉认证最短路径是自动寻找的，避免了认证过程中认证路径的不确定性。

一个带有链路成本和 *CA* 验证成本的网络示例如图 3 所示。任意 2 个 *CA* 之间可以进行双向或单向的交叉认证。假设对于任意 2 个根 *CA*，其证书传递路径长度与传递方向无关。如果将路径总成本定义为此路径上所有路径的传递成本与所有根 *CA* 认证成本的总和，那么一个节点对之间的最短路径即是具有最小路径总成本的路径。

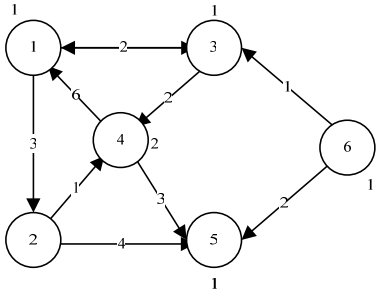

图 3　带有链路成本和 *CA* 验证成本的网络

得到的以节点 1 为根节点的最短路径树如图 4 所示。当算法结束时，它将求得每个节点的最小成本以及最短路径上的下一个节点。

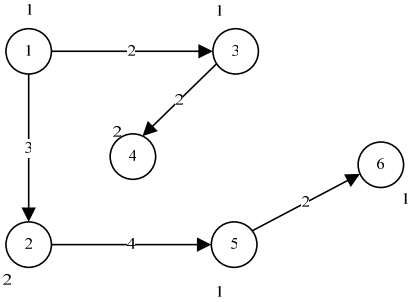

图 4　从节点 1 到其他节点的最短路径树

路径总成本算法的执行过程如表 1 所示。

表 1　路径总成本算法的执行

迭代	*N*	*D*2	*D*3	*D*4	*D*5	*D*6
初始化	{1}	5	3	∞	∞	∞
1	{1,3}	5	3	7	∞	∞
2	{1,2,3}	5	3	7	10	∞
3	{1,2,3,4}	5	3	7	10	∞
4	{1,2,3,4,5}	5	3	7	10	13
5	{1,2,3,4,5,6}	5	3	7	10	13

(1)节点 1 比较它到与它直接相连的各个节点的成本，发现节点 3 是最近的，并且成本为 2+1。如果只需寻找节点 1 到节点 3 的最短路径，则算法到此结束。节点 3 被加入集合 *N*。从节点 1 通过节点 3 到其他节点的新的最小成本会被确定。可以发现，到节点 4 的一条新的最短路径通过节点 3 并且路径总成本为 2+1+2+2=7，一次迭代 1 中的第 3 个表项会发生改变。

(2)选择节点 2 作为离节点 1“次近”的节点并把它加入集合 N。通过连接节点 1 和节点 2，对图进行更新。发现节点 5 通过节点 2 的一条路径总成本为 10。

(3)节点 4 被加入到 N 中，并且通过节点 3 在树图中连接到节点 1。此时没有发现新的最短路径。

(4)发现节点 5 是离节点 1“第 4 近”的节点，它通过节点 2 与节点 1 相连。此时可以发现一条从节点 5 到节点 6、路径总成本为 13 的新的最短路径。

(5)节点 6 以总成本 13 被加入 N。节点 6 通过节点 5 和节点 2 连接到节点 1。至此，寻找到了节点 1 到节点 6 的最短路径，并且从节点 1 到所有节点的最短路径都已找到，算法结束。

3 算法分析

所设计的模型可用以下数据结构表示，其中，设信任路径图中的节点数为 V。

```
//根 CA 的结构体类型
typedef struct
{
 //根 CA 的标志名 Distinguished Name
char nameCA[DNLENGTH];
//搜索服务器地址(编号)
 int searchSvrAddr;
//搜索过程中已建立认证关系的目标 CA 集合(链表)
   CertNode* head;
}CA;
//证书链表节点的结构体类型
typedef struct
{
 //证书接收者的 DN
char recvCADN[DNLENGHT];
//证书颁发者到该节点的传递成本与认证成本之和
int cost;
//已经搜索到的目标根 CA 的认证路径
int savedPath[V];
//链表中的下一个节点 CA
CertNode *next ;
}CertNode;
```

算法中有 3 个一维数组：$s[n]$：保存已求得最短路径终点的集合；$distance[n]$：保存从源点到其余节点之间当前的最短路径长度；$path[n]$：保存从源点到终点所经过的路径[6]。因此，算法的空间复杂度是 $O(n)$。

如果有 n 个节点，算法本身时间复杂度为 $O(n^2)$，再每次以一个节点为源点，重复执行 n 次，得到任意两点间的最短路径，所以，总的时间复杂度为 $O(n^3)$。

利用新的认证算法搜寻最短路径与直接建立认证关系的比较如表 2 所示。

表 2　使用新的认证算法搜寻最短路径与直接建立认证比较

比较项目	使用新的认证算法	直接认证
与新的根 CA 认证的方式	利用已有认证关系，逐级认证	建立新的认证
与新的根 CA 认证的速度	快	慢
建立交叉认证总数	少	多

本算法能够自动寻找网状认证模型中的交叉认证最短路径，避免了现有认证过程中的认证路径的不确定性，并使用了一种较优的路径查找算法。但是，每次发生证书更新或认证关系变化后，都会通知其他根 CA 和搜索服务器，在进行认证时需要重新计算最短路径。以后在没有变化和更新的情况下，可以使用以前的最短路径进行交叉认证，避免了路径搜索和交叉认证的建立。使用新的认证算法与直接建立认证所消耗的时间比较如图 5 所示。

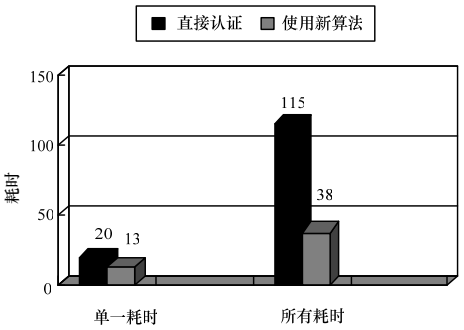

图 5　使用与未使用新的认证算法消耗时间比较

2 个原本没有认证关系的 *CA* 在建立交叉认证之前需要先将自己的信息发到对方的注册中心(Registration Authority, RA)。RA 是接收申请数字证书的部分，它执行注册和鉴别用户证书请求的必要步骤[7]。在 RA 检验了信息(如姓名、电子邮件、物理地址)的真实性后，才可以发放证书，这一过程会消耗较长的时间(注册时间 T_R)，远大于认证时间 T_A，即 $T_R>>T_A$，假设 $T_R=10T_{A(max)}$(实际应该大于这个比例)，则在图 2 的例子中，$T_R=20$, $T_{A(max)}=2$。

4 结束语

本文针对中型 PKI 系统中需要进行一定数量的交叉认证的特点，设计了一种基于 Dijkstra 算法的认证路径搜索算法。树型模型的证书路径容易扩展，但对根 CA 的安全性要求很高；网状型模型具有很好的灵活性，但存在从用户证书到可信任点建立证书的路径不确定的问题。本文所设计的模型弥补了以上 2 种模型的缺点，实现了域间认证路径的有效查找和认证成本的降低，具有较强的实际意义。

参考文献

[1] 杨绚渊, 刘　艳, 陆建德. 一种改进的交叉认证路径构造算法设计[J]. 计算机工程, 2006, 32(24): 146-148.

[2] 叶　敏, 方　勇, 周安民. 公钥基础设施信任模型中信任路径问题研究[J]. 信息与电子工程, 2006, 4(1): 6-9.

[3] Leon-Garcia A, Widjaja I. Fundamental Concepts and Key Architectures[M]. 北京: 清华大学出版社, 2004.

[4] 刘英娜, 徐向阳, 孟　洋. PKI 信任模型研究[J]. 计算机安全, 2006, 6(10): 25-27.

[5] 冯玉翔. PKI 系统交叉认证技术的研究与实现[D]. 广州: 华南理工大学, 2000.

[6] 田鲁怀. 数据结构[M]. 北京: 电子工业出版社, 2006.

[7] Arther W C, White G B. Cothren C, et al. Principles of Computer Security[M]. [S. l.]: The McGraw-Hill Companies, 2004.

编辑　张　帆