一种基于时间域和信任度的分布式证书链搜索算法①

郝晓晓 张卫丰 (南京邮电大学 计算机学院 江苏 南京 210003)

摘 要: 在分布式系统中在分布式环境中，基于目标的分布式证书链搜索算法的提出，找到并检索出所需要的证书，但是目前得信任管理系统还存在以下不足：(1)委托深度没有得到控制；(2)证书的有效期没有得到体现。针对上述问题，提出了一种基于时间域和信任度的分布式证书链搜索算法，并结合具体的例子给出了该算法的使用。

关键词: 信任管理；分布式；证书链；信任域；时间域

A Time Domain and Trustworthiness-Based Distributed Credential Chain Discocery Algoritnm

HAO Xiao-Xiao, ZHANG Wei-Feng

(College of computer, Nanjing University of Posts & Telecommunications, Nanjing 210003, China)

Abstract: In the distributed environment, the goal-directed, distributed chain discovery algorithm finds and retrieves credentials needed. However, the exiting trust management systems have some flaws as follows: (1) delegation depth is not controlled effectively. (2) the validity of the certificate has not been reflected. To address the problems above, this paper proposes a time domain and trustworthiness-based distributed credential chain discovery algorithm, and gives a specific example of the algorithm.

Keywords: trust management; distribution; credential chain; trustworthiness; time domain

2001 年，Li 和 Winsborough 等人对基于角色的信任管理体系 [1](RT，Role-based Trust- management)提出了一种面向目标的分布式证书链发现算法。在该算法中，证书及其含义的集合用证书图(Credential Graph) 表示，图中的节点代表证书上的一个角色表达式，每个证书对应图的一条边，边的终点是角色。证书链发现实际就是路径的发现。

1 引言

1.1 文章安排

本文的组织如下：第一节给出 RT(Role-based Trust-management)基本语法；第二节给出了改进后的 RT0 证书的组成与语义，第三节给出了引入改进后证书图的构建并给出具体实例，第四节证明算法的完备性，第五节结束语。

1.1.1 基本介绍

随着证书的不断更新和撤销，证书的有效期没有得到体现；再者，现有的信任管理系统在进行授权委托[2]过程中实体间的信任度固定不变，对委托的控制深度，每个节点上用一个固定的阀值来界定，这与实际情况不相符。本文在 Li 文中分布式算法基础上，改进了 RT0 证书，引入了时间域；在现有的信任度[3]、信任阀值的基础上进行了改进，使它们随时间动态变化。通过这种改进，在与时间有关的信任管理系统的应用领域，实体之间可以根据证书的有效期，信任度，较为合理、安全的进行证书链的搜索发现[4]。

2 RT0基本语法

RT 使用角色来表示属性，角色定义了一个实体集

① 收稿时间:2009-05-19

合，实体就是这个角色的成员。角色可以看作一种属性，实体是一个角色的成员当且仅当该实体有角色所标识的属性。它结合了基于角色的访问控制(RBAC)和信任管理(TM)系统的优点，更简洁直观。RT0 是 RT 的基础部分，下面给出 RT0 语法的基础。

实体(Entity)：是一个唯一标识。实体可以发布证书、提出访问控制请求。通常用大写字母如 A、B 等表示。

角色名(RoleName)：是一个标识符，用字符串表示，通常用小写字母如 r、r1 等表示。

角色(Role)：　是一个实体后面跟一个角色名如 A.r，B.r 等表示。

链接角色(LinkedRole)：是一个实体后面跟两个角色名，这两个角色名用“.”来链接，如 A.r1.r2 等表示。

角色交集(Intersection)：如 f1∩f2∩…∩fn 其中 fi(1≤i≤n)是一个实体、角色或链接角色。

角色表达式(RoleExpression)：实体，角色，链接角色的统称，通常表示为 e,e1 等。

Role = {A.r | A∈Entity, r∈RoleName}

LinkedRole = {A.r1.r2 | A∈Entity, r1, r2∈RoleName}

Intersection = {f1∩f2∩…∩fn | fi ∈(Entity∪Role∪LinkedRole)}

RoleExpression=Entity ∪ Role ∪ LinkedRole ∪Intersection

RT0 有四种证书，用来定义角色。

① A.r<--B

A 和 B 都是实体，r 是一个角色名。该证书表示 A 定义 B 是 A 的角色 r 的一个成员，即 A 授权 B 拥有属性 A.r，这里 A 和 B 可以为同一实体。

② A.r<--B.r1

A 和 B 都是实体，r 和 r1 是角色名。该证书表示 A 定义它的角色 r 包含 B 的角色 r1 的所有成员实体。换句话说，A 定义角色 B.r1 比角色 A.r 有更大的权利，即 B.r 的成员能够做角色 A.r 被授权的任何事情。即如果 B 授权一个实体拥有属性 B.r1，那么 A 就授权该实体拥有属性 A.r。

③ A.r<--A.r1.r2

A 是一个实体，r、r1 和 r2 都是角色名。形如 A.r1.r2 称为链接角色。该证书表示如果实体 B 是 A.r1 的成员，那么 A 定义它的角色 r 包含 B 的角色 r2 的所有成员。即如果 A 授权 B 拥有属性 A.r1，B 授权某个实体 C 拥有属性 B.r2，那么 A 就授权实体 C 拥有属性 A.r。这是一种基于角色的委托，A 根据 B 的某些属性来决定是否向 B 委托权利，而不是基于 B 的身份。

④ A.r<--f1∩f2∩…∩fn

A 是一个实体，n 是大于 1 的整数，f j(1≤j≤n)是一个实体、一个角色或者一个链接角色。f1∩f2∩…∩fn 称为交集。这个证书表示 A 定义同时为 f1,…,fn 的实体成员是角色 r 的成员。即如果某个实体同时具有 f1,…,fn 代表的属性，那么 A 就授权该实体拥有属性 A.r。

3　改进后的RT0证书组成及语义

3.1 带时间衰减的信任度和信任阀值的更新

信任度[5]通常定义为个体 A 认为个体 B 会按照自己的期望的方式做事的信任程度，这里考虑的是随时间动态变化的情况。实体 A 通过与实体 B 的直接交互经验获得的对 B 的信任的程度的一种量化，通常称为直接信任度(DT)。

每次建立证书时，都要进行一次直接信任度的计算，以供下次使用作为参考。另外直接信任度和间接信任度在大多数情况下跟时间有较大的关系，如果一个证书长时间没有发生更新，很难保障现有的信任状况，为此引入时间衰减系数 ρ 和 ϕ，时间单位 T。直接信任度的计算如下：

$$DT_{i,j,t}^{new} = DT_{i,j,t0}^{old} \times \rho^{(t-t0)/T}$$

信任阀值即为可信的最小信任度，对角色中权限的使用的一种约束，只有当信任度大于此值时才认为可信，小于此值就认为其不可信，不与之交易。在实际应用中可根据实际情况来确定每个结点得初始阀值，本文中 TS(i)初始化为 0.5，最大为 1.0，最小为 0.0，ϕ 是我们根据实际情况规定的一个变化的单位量，一般规定非常小，经过很多次的累积才会对结果产生影响，是一种历史经验的积累，这与现实是相符的。对于证书链的所有中间结点的阀值，证书链构建成功时，分别减小一个很小的值，构建失败时分别增加一个很小的值。

if S(i)-ϕ >=0 then TS(i)= TS(i)-ϕ
else S(i)=0
else if TS(i)+ ϕ >=1.0 then TS(i)=TS(i) +ϕ
then TS(i)=1

信任阀值计算举例：

EPub.discount←EOrg.preferred (1)
EOrg.preferred←StateU.student (2)
StateU.student←RegistrarB.student (3)
RegistrarB.student←Alice (4)

EPub.dicount $\xleftarrow{(1)}$ EOrg.preferred $\xleftarrow{(2)}$ StateU.student $\xleftarrow{(3)}$ RegistrarB.student $\xleftarrow{(4)}$ Alice

从 EPub 开始顺利的建立了证书链，找到 Alice，对结点 Alice、RegistrarB.student、 StateU. Student 、EPub.dicount 的信任阀值更新为：现有的基础上减去ϕ。

3.2 时间域的更新

由于证书的有效期没有体现，现引入时间域的概念，即在某一段时间内证书是有效的，比如[t1,t2]等，其中 t1,t2 是单位时间得累积值。每次证书产生时，对已经存在的更新其时间域，不存在的直接产生带有时间域的证书。时间域计算，[t1,t2]∩[t3, t4] = [max(t1,t3)，min(t2,t4)]，举例如下：

EPub.dicount $\xleftarrow{(7,15)}$ EOrg.preferred (1)
EOrg.preferred $\xleftarrow{(8,13)}$ StateU.student (2)
StateU.student $\xleftarrow{(9,14)}$ RegistrarB.student (3)
RegistrarB.student $\xleftarrow{(6,12)}$ Alice (4)

根据定义可知，(7,15)∩(8,13)=(8,13)…依次计算，最后得到，在时间域(9,12)内，Alice 可以享受 EPub 的打折优惠。

3.3 带有时间域和改进信任度的证书

通过以上讨论，改进后证书的格式为(R,S,T,DT,TS)，其中：

R 是一个角色；S 是一个角色表达式；T 是证书的有效期，以区间的形式给出得；DT 是在时间因素影响下得直接信任度；TS 是信任阀值。

改进后的证书可以写成 R $\xleftarrow{(\alpha,t)}$ S,其有以下四种类型的证书。

类型 1：A.r $\xleftarrow{(\alpha,t)}$ B，其中 A 和 B 是实体名，r 是角色名，α 是信任度，t 是证书有效期，其含义是，实体 A 定义实体 B 在证书有效期 t 内，以信任度成为角色 A.r 的一个成员。

类型 2：A.r $\xleftarrow{(\alpha,t)}$ B.r1，其中 A 和 B 是两个不同的实体名，r 和 r1 是角色名，α 是信任度，t 是证书有效期，其含义是，实体 A 定义所有 B.r1 的成员，在证书有效期 t 内，以信任度成为角色 A.r 的成员。

类型 3：A.r $\xleftarrow{(\alpha,t)}$ A.r1.r2，其中 A 是实体名，r 和 r1 是角色名，A.r1.r2 是链接角色，α 是信任度，t 是证书有效期，其含义是，如果实体 B 具有角色 A.r1，且 B 又定义 C 具有 B.r2 角色，则在证书有效期 t 内，以信任度成为 A.r 的成员。

类型 4：A.r $\xleftarrow{(\alpha,t)}$ f1∩f2∩…∩fn ，其中 fi 是一个实体、角色或以 A 开头的链接角色，f1∩f2∩…∩fn 称为角色交集，其含义是，A 定义同时具有所有 fi 的成员身份，在证书有效期 t 内，以信任度α 称为角色 A.r 的成员。

4 证书图的构建及相应的实例

在信任管理[6]中处理实体所提出的使用角色的请求，则要判断请求者是否具有拥有此角色的权限，即实体是否为资源提供者所定义的角色中的成员。在处理授权请求时，通常要解决三种查询[7]：

① 给出一个角色表达式 e，确定其成员集合 expr[Sc](e)，对应后向搜索；

② 给出一个实体 D，确定所有包含实体 D 的角色，对应前向算法；

③ 给定一个角色表达式 e 和一个实体一个成员 D，判断 D∈expr[Sc](e)，即判断实体 D 是否为角色 e 中的，对应双向算法。

其中：Sc 是将角色映射到实体的函数；[Sc](e)是角色表达式 e 所包含的实体；expr[Sc](e)是角色表达式 e 所包含的各类成员的集合。

证书图(带权有向图)中，C 为证书集合；Entities(C)表示 C 中的实体集合；Name(C)表示 C 中的角色名集合；Intersection(C)表示 C 中的角色交集集合；FExps(C)表示 C 中的非交集的角色表达式。C 上的每个证书图实际上就是一个有限角色表达式集合 Q⊆角色表达式的参数化，证书图用 $G_{C:Q}$，结点集 $N_{C:Q}$，边集合 $E_{C:Q}$，$N_{C:Q}$ = FExps(C)∪Intersection(C)∪Q，实体、角色以及其它的角色表达式转化成结点，证书转化成边，每个结点的信任度初始值设为 0.5，然后每次在构建证书图时更新信任度及信任阀值。

下面给出国内 ACM 计算机科学文献库场景中用到的证书链，数据库在国内隶属于中国科学院科学数字图书馆(CSDL)，该文献机构通过委托证书允许某些已经与其结成联盟关系大学或机构向其推荐新的大学或机构。

ACM.ordinary $\xleftarrow{(1.0,(1,20))}$ CSDL.member //总部注册会员可以使用 ACM 的服务

ACM.ordinary $\xleftarrow{(1.0,(1,18))}$ CSDL.member ∩ ACM.ally.student //联盟大学的学生同时又是总部的会员可以最优先的服务

ACM.ally $\xleftarrow{(0.96,(2,19))}$ NJU //南京大学是联盟大学

ACM.ally $\xleftarrow{(0.9,(3,16))}$ NJU.recommended //委托南京大学推荐新大学进入联盟的权力

NJU.recommended $\xleftarrow{(0.8,(3,20))}$ SEU //南京大学推荐东南大学进入联盟

NJU.recommended $\xleftarrow{(0.85,(2,15))}$ SEU.recommended //南京大学委托东南大学推荐新大学进入联盟的权力

NJU.student $\xleftarrow{(1.0,(1,12))}$ A //A 是南大学生

SEU.recommended $\xleftarrow{(0.84,(2,25))}$ NUPT //东南大学推荐南京邮电大学进入联盟的权力

SEU. student $\xleftarrow{(1.0,(1,15))}$ B //B 是东大学生

NUPT. student $\xleftarrow{(1.0,(2,21))}$ C //C 是南邮学生

CSDL.member $\xleftarrow{(0.95,(2,30))}$ A //A是中科院数字图书馆成员

CSDL.member $\xleftarrow{(0.95,(2,30))}$ B //B 是中科院数字图书馆成员

CSDL.member $\xleftarrow{(0.95,(2,30))}$ C //C是中科院数字图书馆成员

假设 e 是实体，r 是角色或链接角色或角色交集，在 e 取得 r 及 r 的信任度和有效期的过程中，可能包含 3 种类型的边：

(1) A.r $\xleftarrow{(\alpha,t)}$ e(实体、角色、链接角色、角色交集)∈ E C:Q 直接对应证书图中的边。

(2) f1∩f2∩…∩fn $\xleftarrow{(\alpha,t)}$ e(实体)∈ $E_{C:Q}$ 并称此边为衍生边，所有路径{fj $\xleftarrow{*}$ e(实体)|j∈ [1,k]}为该边的一个支持集，该边是由它的支持集中的路径导出的，因此取这些路径中信任度最小值作为其信任度，时间域的最小交集作为其时间域。

(3) A.r1.r2 $\xleftarrow{(\alpha,t)}$ e∈ $E_{C:Q}$ 并称此边为衍生边，A.r1 $\xleftarrow{(\alpha,t)}$ e 称为该边的支持集，信任度和时间域等于 A.r1 $\xleftarrow{(\alpha,t)}$ e 的信任度和时间域。

其中(2)、(3)类型得边是由其支持集导出得，支持集中可能还包含此类型的边，最终会归结到直接对应于在有效时间域内证书图中的边的信任度[8]，所以其计算过程可能是递归的。通过上述步骤构建的证书图如：

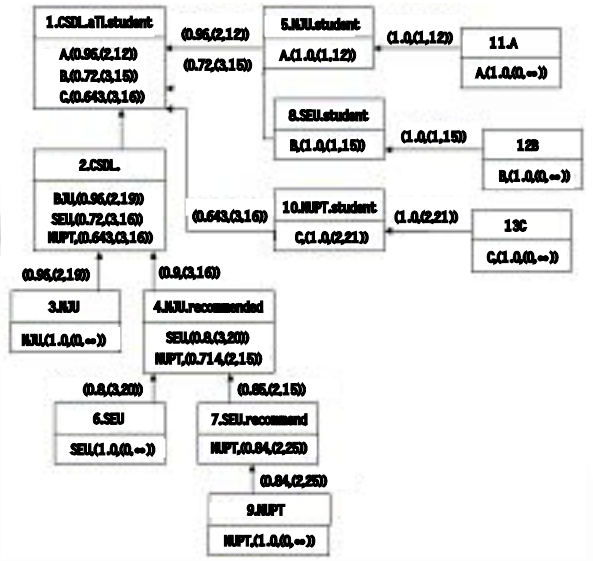

图 1　构建的证书图的过程

5　算法可行性证明

如何保证实际存在的路径一定能够通过本算法得到，下面给出算法合理性的证明：

如果结点 e 在证书图中有一实体 B 作为它的一个后向解，则存在一条路径 e $\xleftarrow{(\alpha,t)}$ B，如果存在路径 e $\xleftarrow{(\alpha,t)}$ B，则在证书图中 B 为结点 e 得一个后向解，并且满足路径中的信任度和时间域。

本文采用归纳法证明，实体作为后向解与路径的一一对应关系，实体作为前向解时类似。

如果在路径中存在直接路径 e2 $\xleftarrow{(\alpha,t)}$ e1，在算法中直接增加结点 e1 结论成立。

假设 ei $\xleftarrow{(\alpha,t)}$ ek 成立，算法通过判断信任阀值和有效期得到 ei 的解是否为 ek 的解，再对 ek 进行后向查找解的集合，因此算法得到路径 ei $\xleftarrow{(\alpha,t)}$ ej，即如果存在授权路径，通过算法总能得到正确的结果。

假设 N 为证书集中的证书个数，M 为证书中的大小 size(c)=$\sum_{A.r\xleftarrow{(\alpha,t)}e\in C}|e|$且 |A|=|A.r|= |A,r1, r2|=1，|f1∩f2∩…∩fk|=k,可以看出在满足信任阀

(下转第 146 页)

(上接第 142 页)

值和有效期得情况下，与 LI 提出的算法得时间复杂度相同，即最大时间复杂度为 O(n^3 +mn)空间复杂度为 O(mn)。

6 RT0基本语法

在 RT0 的基础上提出了基于证书时间域和改进的信任度证书链搜索算法，将时间考虑到了证书和信任度的更新中，并在此基础上对信任的传递深度通过信任阀值进行了控制，符合信任得主观性。随着证书内容的增多，比如证书有效性的增加，如何存储更有效；对带有参数的基于角色的访问控制[9]等都需要进一步研究。

参考文献

1 Li NH, Winsborough WH, Mitchell JC. Distributed credential chain discovery in trust management (full version). New York: ACM Press, 2001.156－165.

2 张志勇,黄涛.信任管理中基于角色的委托授权研究进展.计算机应用研究, 2008,25(6):1601－1605.

3 陈博,李明楚,任一支.基于信任度的改进凭证连发现算法.计算机工程, 2008,34(22):174－176.

4 祝胜林,杨波,张明武.信任管理中证书链发现的研究.计算机工程与应用, 2007,43(8):111－113.

5 廖俊国,洪帆,朱更明,杨秋伟.基于信任度的授权委托模型.计算机学报, 2006,29(8):1265－1270.

6 刘鹏,刘欣,陈钟.信任管理综述.计算机工程与应用, 2004,32:38－43.

7 Li NH, Winsborough WH, Mitchell JC. Distributed Credential Chain Discovery in Trust Management. Journal of Computer Security, 2003, 11(1):35－86.

8 程男男,杨波.一种带有信任度的基于角色的信任管理模型.计算机应用研究, 2006,23(1):100－102.

9 Zhu X, Wang S, Hong F, et al. Distributed credential chain discovery in trust-management with parameterizeed roles. Chinese Journal of Computers, 2006,29(8):1266－1270.