第32卷 第21期 计 算 机 工 程 2006年11月
Vol.32 No.21 Computer Engineering November 2006
·安全技术· 文章编号：1000—3428(2006)21—0175—02 文献标识码：A 中图分类号：TP391

# 基于图论的一致性验证算法

饶懿璇，杨 波
(计算机网络与信息安全教育部重点实验室，西安 710071)

**摘 要：**提出了 PolicyMaker 信任管理系统的一种新的一致性验证算法，此算法运用图论中的深度优先遍历理论以及图的动态特性，在遇到撤消授权关系的凭证时，将其对应的边删除，然后重新搜索，寻找其他证书链直至搜索结束。它解决了原算法中否定安全凭证的问题，并且通过与原算法时间复杂度与空间复杂度的比较，证明此算法更加简单快捷。

**关键词：**信任管理；否定安全凭证；图论

# A Graphic-based Theory Proof of Compliance Algorithm

RAO Yixuan, YANG Bo
(Key Lab of Computer Network and Information Security, Ministry of Education, Xi'an 710071)

**【Abstract】**The paper puts forward a new proof of compliance algorithm in PolicyMaker, the algorithm applies graphic theory's DFS notion and the dynamic character, in the face of withdrawing the delegation, deletes the delegation side, then newly searches, finds other chain discovery until the end. It solves the negative credentials, compared with the time and space complicated degree of the original arithmetic, the algorithm is more simple and shortcut.

**【Key words】** Trust management; Negative credential; Graphic

随着 Internet 的快速发展，出现许多新的大规模、开放的分布式系统。这些系统从面向封闭的、熟识用户群体的和相对静态的形式向开放的、公共可访问的和高度动态的服务模式转变。这种转变使传统的访问控制机构不再适用解决 Web 的安全问题。

因此，Web 安全需要新的思想和方法。M.Blaze 等为解决 Internet 网络服务的安全问题首次使用了“信任管理”一词，其基本思想是承认开放系统中安全信息的不完整，系统的安全决策需要依靠可信任第三方提供附加的安全信息。信任管理通过定义表示授权和访问控制的语言及提供信任管理引擎，来解决访问控制中请求是否被允许的问题。

## 1 信任管理系统

PolicyMaker 是 M.Blaze 等提出的信任管理思想实现的信任管理系统。它采用一种完全可编程的机制来描述安全策略和安全凭证，所有的安全策略和安全凭证均可归结为断言。每个断言可表达为一个 $(f,s)$ 对，其中 $s$ 表示权威源(source of authority)，$f$ 则是一段用于描述实际授权内容或委托授权内容的程序。在一个安全策略断言中，$s$ 被赋予关键字 policy。策略断言是信任管理引擎必不可少的输入参数，描述了应用系统判断请求是否可接受的最终权威根源。而在一个安全凭证断言中，$s$ 被赋予该安全凭证签发者的公钥。在采纳某个安全凭证时，公钥用于验证该安全凭证的可靠性。

由于 PolicyMaker 没有指明特定的断言描述语言，因此其策略一致性证明验证算法必须独立于特定的断言描述语言。PolicyMaker 进行一致性证明验证的一般步骤描述如下：首先建立一个仅包含请求字符串 $r$ 的黑板。随后，调用断言，同一断言可以根据需要调用多次，另外断言由本地运行环境解释。当断言 $(f_i,s_i)$ 运行时，先读黑板中的内容并根据内容加入一条或多条接受记录 $(i,s_i,R_{ij})$，但不能够删除其他断言已写入黑板的接受记录。其中 $R_{ij}$ 表示一个被权威源 $S_i$ 所证明的特定应用操作，可以是一个输入请求 $r$，也可以是一些用于断言间交互的操作。算法本身不需要理解和处理 $R_{ij}$，面向特定应用的断言程序 $f_i$ 处理 $R_{ij}$。最终，所有断言调用完毕。若黑板中存在一条能证明请求 $r$ 的接受记录，则一致性证明验证成功。

## 2 一致性验证算法

本文运用图中深度优先遍历的理论，提出了一种新的一致性验证算法，解决了否定安全凭证的问题，并且提高了效率。其中，图中的点(prinical)代表授予权限或接受权限的点，图中的有向边(authorization)代表授权的内容。

算法如下：

(1)将发出请求 r 的个体存储于图中(不失一般性，假设用邻接表存储)。

(2)调用断言，如果有包含请求 r 的断言，将其记录于图中。

(3)在图中，从请求点 $u_0$ 出发沿它的反方向，用深度优先遍历的方法寻找一条通向 policy 的路径。如果有则输出成功，如果不存在则继续增加断言，直到所有断言调用完，还未成功，则输出不成功。

具体搜索算法如下：

(1)建立一个集合 $u=\{u_0\}$ 以及 $v=\{u_0\}$。

(2)从 $u_0$ 出发，在所有指向它的边中，选中一条，将选中那条边的另一个邻接点与 $u$ 中的点比较，如有相同的点，将其删除，如无，将其添加到 $u$ 和 $v$ 中，且比较此点是否与 *policy* 相同，如是，则输出成功，如不是，则转到(3)。

**基金项目：**国家自然科学基金资助项目(60573043,60372046)；现代通信国家重点实验室基金资助项目(51436040204DZ0102)
**作者简介：**饶懿璇(1982－)，女，硕士生，主研方向：网络安全；杨 波，教授、博导
**收稿日期：**2005-11-24 **E-mail：**joyce552211@163.com

(3)将找到的这条边做记号，并从找到的这个点(设为 $u_i$ )出发，重复(2)。如果向下继续搜索需要条件，譬如，需要 $u_2$ 同时满足授权给某个点时，就要搜索 $u_2$ 看其是否授权即可，然后判断 $u_i$ 是否能继续向下进行下去。直到 $u_i$ 没有指向它的边为止，将 $u_i$ 从 $u$ 中删除，然后返回上一个点，看是否有其它路径，如有，则继续搜索，如无，则将那个点继续删除，返回直至 $u_0$ 为止。

(4)重新从 $u_0$ 指向它的边中选择一条未做标记的边，重复(2)、(3)，一旦发现有记号的路，则放弃选择另一条继续。

(5)直到指向 $u_0$ 的路都有标记为止。

具体实现，如图 1 所示。

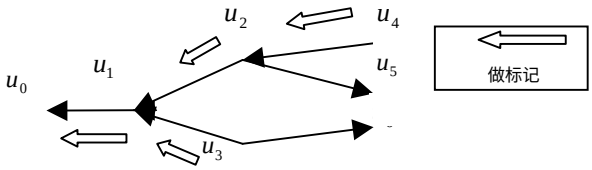



**图 1 搜索不成功**

搜索过程(图 2)如下：

$\{u_0\} \to \{u_0,u_1\} \to \{u_0,u_1,u_2\} \to \{u_0,u_1,u_2,u_4\} \to \{u_0,u_1,u_2\} \to \{u_0,u_1\} \to \{u_0,u_1,u_3\} \to \{u_0,u_1\} \to \{u_0\}$

最后输出不成功。

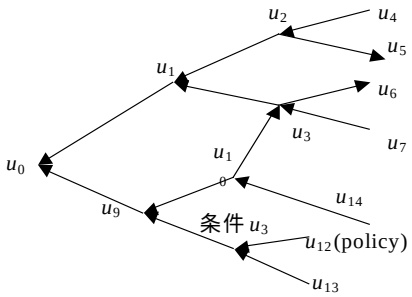



**图 2 搜索成功**

$\{u_0\} \to \{u_0,u_1\} \to \{u_0,u_1,u_2\} \to \{u_0,u_1,u_2,u_4\} \to \{u_0,u_1,u_2\} \to \{u_0,u_1\} \to \{u_0,u_1,u_3\} \to \{u_0,u_1,u_3,u_7\} \to \{u_0,u_1,u_3\} \to \{u_0,u_1,u_3,u_{10}\} \to \{u_0,u_1,u_3,u_{10},u_{14}\} \to \{u_0,u_1,u_3,u_{10}\} \to \{u_0,u_1,u_3\} \to \{u_0,u_1\} \to \{u_0\} \to \{u_0,u_9\} \to \{u_0,u_9,u_{10}\} \to \{u_0,u_9\} \to \{u_0,u_9,u_{11}\} \to \{u_0,u_9,u_{11},u_{12}\}$

最后输出成功。

## 3 算法的比较

令n为顶点数，e为边数，此算法的时间复杂度为$O(n^3)$，因为在最坏情况，就是每次要搜索所有的边，其时间复杂度为$O(n^2)$，在假设每个点都带条件的话，就是$O(n^2\times n)$，即时间复杂度为$O(n^3)$。而原算法最终的时间复杂度为$O(mn^2(mns)^c)$,所以比原算法要快,并且空间复杂度为O(n+2e)也很小，比原算法提高了效率。

对于整个过程，每读入一条断言都会使图中的点以及授权关系动态地改变。所以每当输入否定凭证的断言时，原算法仅能处理满足单调性的策略断言，不能删除其他断言已写入黑板的接收记录，而此新算法只需将对应的授权关系的边删除，然后重新搜索，解决了否定安全凭证的问题。

存储图中的点与边可以用 2 种方式来存储：邻接矩阵和邻接表。当图为稠密图时用邻接矩阵；而当图为稀疏图时，用邻接矩阵存储空间浪费很大，所以用邻接表更好。对上述算法，如与请求 r 一致的凭证很多时，就用邻接矩阵，将其存储空间设为足够大。如与请求 r 一致的凭证很少时，用邻接多重表，将点、邻接表和逆邻接表一起记录下来。

## 4 结束语

本文介绍了有关信任管理模型的基本概念，提出了的新的一致性验证算法，解决了不可撤消性，并使其更加简便快捷。但该算法还有一些不足，比如没有将所有可能出现的情况考虑到算法中去以及难以处理不确定的安全信息等问题。

### 参考文献


1 Blaze M, Feigenbaum J, Lacy J. Decentralized Trust Management[C]. Proc. of the 17th Symposium on Security and Privacy, 1996: 164-173.

2 Blaze M, Feigenbaum J, Strauss M. Compliance Checking in the PolicyMaker Trust Management System[C]. Proc. of the Financial Cryptography'98, 1998: 254-274.

3 Weeks S. Understanding Trust Management Systems[Z]. http://citeseer.ist.psu.edu/weeks01.understanding.html.

4 徐 锋, 吕 建. Web 安全中的信任管理研究与进展[J]. 软件学报, 2002, 11(13): 2057-2064.

5 张选平, 雷咏梅. 数据结构[M]. 西安: 西安电子科技大学出版社, 2002.


（上接第 174 页）

$u(x) = x^7 + x^4 + x^3 + x^2 + 1$； $u(x) = x^7 + x^6 + x^3 + x^2 + 1$；

$u(x) = x^7 + x^6 + x^4 + x^3 + 1$； $u(x) = x^7 + x^6 + x^5 + x^2 + x$；

$u(x) = x^7 + x^6 + x^4 + x^3 + x^2$；

## 4 结束语

本文通过这种方法构造出了密码性能与 AES 的 $S$ 盒相当的 8×8 的 $S$ 盒。由于在构造过程中都使用了求逆变换，因此构造出来的一批 S 盒的非线性度都为 112。无论仿射变换怎样改变都不影响 $S$ 盒的非线性度。同样，实验表明，用构造 AES $S$ 盒的方法所构造出来的其他 $S$ 盒的差分均匀度也都是 4/256。惟一有变化的是 $S$ 盒的输出比特的雪崩概率。

### 参考文献


1 冯登国, 吴文玲. 分组密码的设计与分析[M]. 北京: 清华大学出版社, 2000.

2 刘晓晨, 冯登国. 满足若干密码学性质的 S-盒的构造[J]. 软件学报, 2000, 11(10): 1299-1302.

3 谷大武, 徐胜波. 高级加密标准(AES)算法——Rijndael 的设计[M]. 北京: 清华大学出版社, 2003.

4 Daemen J, Rijmen V. AES Proposal: Rijndael: Version 2[EB/OL]. http://www.nist.gov/aes, 1999-09-03.

5 陈 勤, 周 律. Rijndael 分组密码与差分攻击[J]. 小型微型计算机系统, 2003, 24(4): 676-679.

6 师 军, 张福泰, 王耀燕. 高级加密标准Rijndael算法中的S盒及其实现[J]. 小型微型计算机系统, 2003, 24(7): 1207-1209.

7 张玉安, 冯登国. RIJNDAEL 算法 $S$ 盒的等价生成[J]. 计算机学报, 2004, 27(12): 1593-1600.

8 阮传概, 孙 伟. 近世代数及其应用[M]. 北京: 北京邮电大学出版社, 2001.