第 35 卷 第 1 期 计 算 机 工 程 2009 年 1 月
Vol.35 No.1 Computer Engineering January 2009
·软件技术与数据库· 文章编号：1000—3428(2009)01—0082—02 文献标识码：A 中图分类号：N945

# 采用信任管理的分布式文件系统 TrustFs

张子鹏，刘海涛，管海兵
(上海交通大学计算机科学与工程系，上海 200240)

**摘 要：**在传统的分布式文件系统中用户无法判断文件的可信任性，针对此问题提出采用信任管理的分布式文件系统 TrustFs，使用数字签名对文件的发布者进行认证，通过信任管理技术评估发布者的可信任度，从而达到帮助用户识别不安全文件的目的。TrustFs 使用可堆叠文件系统的技术实现，可以移植到所有的 Unix 系统，并具有良好的扩展性。

**关键词：**分布式文件系统；信任管理；可堆叠文件系统

# TrustFs: Distributed File System with Trust Management

ZHANG Zi-peng, LIU Hai-tao, GUAN Hai-bing
(Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240)

**【Abstract】**None of the classic distributed file systems can help users to identify which files are trustable. To solve this problem, this paper presents a distributed file system named TrustFs, which uses digital signature for file owner's authentication and trust management techquines for users' trust metric evaluation. Aided by TrustFs, users can easily distinguish unsafe files from others. TrustFs is implemented as a stackable file system, so it can be ported to any Unix system and it is very extensible.

**【Key words】**distributed file system; trust management; stackable file system

## 1 概述

在一个大规模的分布式文件系统中，可能存在某些恶意用户故意共享一些并不安全的文件，如带病毒的程序或者欺骗性的文档。因此，当其他用户打开一个共享文件之前，就必须判断该文件是否是可信任的，可以通过回答 2 个问题来做出判断：(1)这个文件是哪个用户发布共享的；(2)发布这个文件的用户是否可以信任。如果知道文件 A 是由用户 Bob 发布的，同时知道 Bob 是个值得信任的用户，就有理由相信文件 A 是安全的。然而，现有的分布式文件系统并不能有效地给出上面 2 个问题的答案。本文提出了分布式文件系统 TrustFs，通过使用数字签名解决了上文提到的第 1 个问题，并且通过引入信任管理机制解决了第 2 个问题。

## 2 技术背景

### 2.1 信任管理

在一个多用户的系统中，信任管理是用来判断某个用户行为是否可信任的机制。比如，文件系统可以根据 ACL 来进行访问控制，实质上这就是一种基于特定策略的信任管理机制。此外，信任管理系统也可以模拟现实社会中信誉的概念，使用基于信誉的管理方式。现实中的人在与其他人的交互过程中，会逐渐形成对每个人信誉高低的评估，信任管理系统可以模拟这种社会活动，用一个叫做可信任度(trust metric)的值来表示用户之间的信任程度。它根据用户过去在分布式系统中的交互行为，通过收集用户的反馈信息来计算这个值。基于信誉的信任管理机制在大规模的分布式系统中尤其重要，因为系统中的用户众多，交互的双方往往并不认识，没有办法通过一些类似 ACL 的策略来判断对方是否可信。因此，在分布式文件系统中引入基于信誉的信任管理是非常重要的。

### 2.2 可堆叠文件系统

可堆叠文件系统是一种增量式的文件系统开发技术，它实现了 Unix 的文件系统接口，可以像普通文件系统一样挂载在 VFS 之下工作，但本身并不提供基本的文件数据存储，而是依赖底层的文件系统来实现的。可堆叠文件系统的基本工作原理是：接收文件系统相关的系统调用，对参数进行相关处理，然后转发到下层文件系统获取返回结果，进一步处理并返回最终结果。可堆叠文件系统可以在大部分 Unix 系统中实现，有很好的可移植性，文献[1]讨论了它的实现方式；同时它具有灵活性的特点，适合在现有的文件系统之上增加一些特殊的功能。

### 2.3 相关工作

信任管理机制的作用受到越来越多的关注，它能够解决传统的安全手段在一个开放和动态的分布式系统中的不足，在 email, P2P 和语义网系统中已经有较多的应用。

Konfidi[2]的目标是评估email系统中邮件的可信任性。它的结构与TrustFs类似，采用数字签名来保证邮件无法伪造，用一个信任网络来评估用户的信誉。不同的是，它使用的PKI是PGP的web-of-trust机制，而且它的信誉评估比较复杂。

eBay 使用信任管理系统收集每次交易用户双方相互的评价反馈，然后根据这些积累的反馈信息，全局地评估用户的信誉，并公开其可信任度。一个用户的可信任度越高，通常意味着和他交易越安全。

文献[3]讨论了如何在一个 P2P 文件共享系统里，使用信

**基金项目：**国家自然科学基金资助项目(60503013)
**作者简介：**张子鹏(1984－)，男，硕士，主研方向：文件系统；刘海涛，讲师、博士；管海兵，教授、博士生导师
**收稿日期：**2008-06-29 **E-mail：**zpzhang@sjtu.edu.cn

任管理来评估各个节点的可信任性，从而帮助用户识别潜在的恶意用户。

可堆叠文件系统是文件系统开发的重要技术，它常用来增强系统的安全性和性能。

I3FS[4]是一个可堆叠文件系统，它提供了文件的完整性检查和入侵侦测功能。

RAIF[5]是一个使用可堆叠文件系统技术实现的类似RAID的存储系统，它可以增强分布存储的健壮性，也可以通过并行存取提高性能。

## 3 TrustFs 设计与实现

### 3.1 TrustFs 系统架构

TrustFs 的目标是在分布式文件系统中引入信任管理机制，由于可堆叠文件系统非常适合在现有的文件系统之上增加一些特殊功能，因此 TrustFs 是以可堆叠文件系统的形式实现的。TrustFs 利用下层的分布式文件系统(如 NFS 或 Coda)实现基本的文件系统操作，并在打开文件的系统调用中进行信任检查工作。

TrustFs 采用数字签名和 PKI 机制来解决文件发布者的认证问题。用户要发布(共享)一个文件时，必须附上用自己的私钥对文件内容的签名和自己的证书。另一个用户打开这个文件时，只需要通过证书获得发布者的公钥，并验证数字签名的正确性，即可验证发布者的身份。在不知道私钥的情况下，即使有对手攻破文件系统服务器，或者修改网络中传输的数据，他也无法伪造文件的数字签名。同时，如果对手希望通过修改文件内容来进行攻击，检查数字签名可以起到完整性检查的作用。TrustFs 需要与 CA 通信来验证证书的有效性。

TrustFs 中仅知道用户的名字并不足以判断其可信任程度，因此，依赖一个信任管理服务器提供可信任信息，它采取下文中描述的方法，通过用户的反馈来计算全局的可信任度。TrustFs 通过用户的可信任度来判断该用户是否可信。

综上所述，TrustFs 的系统架构如图 1 所示。

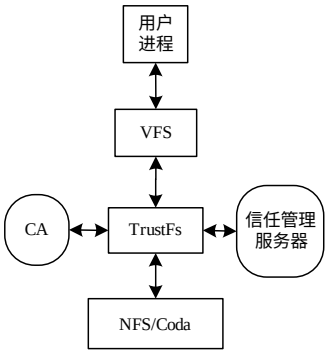


**图 1 TrustFs 系统架构**

### 3.2 TrustFs 文件组织

由于 TrustFs 采用数字签名和 PKI 机制来解决文件发布者的认证问题，因此 TrustFs 需要保存文件的元数据，包括文件发布者的证书以及发布者使用自己的私钥对文件内容的数字签名。其中证书采用常见的 X.509 格式，数字签名采用 RSA 算法，Hash 算法采用 SHA 算法。可能的实现方式如下：

(1)使用数据库来存储证书和数字签名。文献[4]中详细讨论了这种实现方式的优势，但这种方式更适合于本地文件系统。对于分布式文件系统来说，与远程的数据库通信需要额外的端口和延时，并且数据库的部署和维护都增加了系统的复杂性。

(2)添加文件头的方式存储证书和数字签名。利用底层文件系统来存储 TrustFs 的元数据，这种方式易于实现，有较好的可移植性。同时可以利用分布式文件系统在客户端的 cache 机制，有助于提高性能。使用文件头使文件大小和原始数据的位置发生了变化，因此需要对读写等操作进行特殊处理。

综合比较后，TrustFs 采用在原先文件前添加文件头的方式存储元数据，如图 2 所示。TrustFs 打开一个文件详细的过程如图 3 所示。

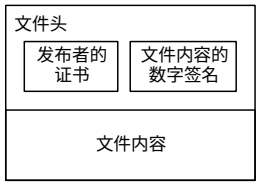


**图 2 TrustFs 的文件结构**

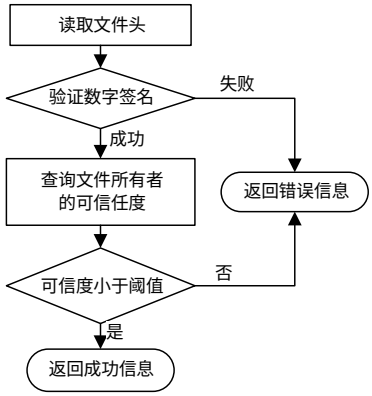


**图 3 文件打开流程**

TrustFs 中文件头对用户是透明的，即用户看到的文件只包含原始的数据。TrustFs 对于文件的操作请求将在可堆叠文件系统中进行偏移量的重新计算，如用户请求读取数据的偏移量为 $n$，TrustFs 发送到下层请求的偏移量应该为 $n+len$，其中 $len$ 为该文件文件头的长度。

TrustFs 只允许文件的发布者(即创建者)修改此文件。因为文件修改后，数字签名要重新计算，而只有文件的发布者才能计算数字签名。如果应用需要多个用户修改同一个文件，那么这些用户需要使用同一个私钥。一种更理想的解决方案是使用群签名(group signature)的技术，但由于其在应用方面还不够成熟，TrustFs 并不支持。

### 3.3 TrustFs 信任管理机制

TrustFs采取文献[3]中用于P2P网络的信任模型来计算可信任度。用户$i$把他对用户$j$的信任评价$s_{ij}$发送给信任管理服务器，其中，$s_{ij}$是一个整数，值越大表示$i$对$j$的评价　　越高。

在文献[3]中，$s_{ij}$被定义为用户$i$对用户$j$满意的交互次数与不满意的交互次数之差；在TrustFs中，当用户$i$打开用户$j$共享的某个文件时，如果这个文件是安全的，则发生了一次满意的交互；如果文件是恶意的，则发生了一次不满意的交互。系统会对$s_{ij}$进行正规化，得到$c_{ij}$：

$$c_{ij}=\frac{\max(s_{ij},0)}{\sum_j \max(s_{ij},0)}$$

（下转第 86 页）