

# 一种基于反馈的信任生成算法

赵晓孔<sup>1,2</sup>, 罗永龙<sup>1,2</sup>, 程超<sup>1,2</sup>, 周正珍<sup>1,2</sup>

(1. 安徽师范大学 数学计算机科学学院, 安徽 芜湖 241003;

2. 安徽师范大学 网络与信息安全工程技术研究中心, 安徽 芜湖 241003)

**摘要:**安全多方计算中一般合作计算者之间互不信任, 为保护私有信息通常采用零信息泄漏的安全策略, 从而增加了合作计算复杂性和通信复杂性。提出了一种基于反馈的信任生成算法, 通过参与者的历史行为评估其信任度, 将参与者的信任度量化表示。根据参与者的信任度采取相应级别的安全措施, 可降低合作的计算复杂性和通信复杂性, 隔离信任值低的恶意参与者。仿真实验结果表明, 该算法在安全多方计算环境中可有效遏制恶意参与者, 提高合作成功率。

**关键词:**安全多方计算; 信任; 算法

中图分类号: TP309

文献标识码: A

文章编号: 1673-629X(2010)12-0166-04

## A Trust Generating Algorithm Based on Feedback

ZHAO Xiao-kong<sup>1,2</sup>, LUO Yong-long<sup>1,2</sup>, CHENG Chao<sup>1,2</sup>, ZHOU Zheng-zhen<sup>1,2</sup>

(1. College of Mathematics and Computer Science, Anhui Normal University, Wuhu 241003, China;

2. Eng. Tech. Research Center of Network and Info. Security of Anhui Normal Univ., Wuhu 241003, China)

**Abstract:** In secure multi-party computation, all cooperation participants are usually regarded distrust, in order to protect private information usually use zero information leakage security policy. This security policy greatly increases computational complexity and communication complexity. Propose a trust generating algorithm, through participants' history behaviors to assess their trust, express trust value in a quantitative way. Taking security policy according to participants' trust value can greatly reduce the cooperation computing complexity and insulate malicious participants. Experimental results show that this algorithm can effectively be against malicious participants and increase the ratio of success cooperation.

**Key words:** secure multi-party computation; trust; algorithm

## 0 引言

自 M. Blaze 等人<sup>[1]</sup>于 1996 年首次提出信任管理概念以来, 信任管理已经得到了广泛研究, 形成了许多基于反馈的信任生成算法<sup>[2~8]</sup>, 这些算法采用多种数学手段和工具在计算精度和复杂度等方面各有所长, 但也都存在一定不足<sup>[9]</sup>, EigenTrust 信任模型<sup>[2,3]</sup>中信任生成算法需预设一个亚可信节点集合; PeerTrust 信任模型<sup>[4]</sup>中信任生成算法存在迭代收敛性问题; Beth 信任模型<sup>[5]</sup>中信任综合计算采用简单的算术平均, 不能有效防止各种针对系统本身的欺骗和攻击; Josang 信任模型<sup>[7,8]</sup>引入了事实空间和观念空间来描述信任

度, 提出了主观逻辑运算符用于信任度的综合计算, 但与 Beth 模型一样不能有效消除针对系统的欺骗和攻击行为。

安全多方计算 (Secure Multi-party Computation, 简称 SMC) 是研究一组互不信任的参与者之间保护私有信息的合作计算问题<sup>[10]</sup>。已有的工作<sup>[11~17]</sup>主要是基于零信息泄露的安全策略, 采用这种安全策略在实际中往往会大幅增加网络资源开销, 降低计算效率。文中针对安全多方计算环境提出一种基于反馈的慢增长快下降信任生成算法 (Slow grow and Fast down value generating Algorithm based on Feedback, 简称 SFAF), 该算法信任值增长缓慢, 但只要有一次合作计算失败信任值就会迅速下降, 不诚实计算次数越多, 信任值下降比例越大。

文中的信任生成算法与传统的几种信任生成算法相比可有效遏制参与者的恶意行为, 防止网络中各种吹捧和诋毁行为。根据信任值采取相应级别的安全措施, 可大幅降低合作计算的复杂度, 提高合作计算成功

收稿日期: 2010-04-13; 修回日期: 2010-07-15

基金项目: 国家自然科学基金项目 (60703071, 60773114); 安徽省优秀青年科技基金项目 (08040106806); 安徽省自然科学基金项目 (070412043); 安徽高校省级自然科学研究重点项目 (KJ2010A133)

作者简介: 赵晓孔 (1984-), 男, 安徽六安人, 硕士研究生, 研究方向为信息安全、可信计算; 罗永龙, 博士, 教授, 博士生导师, 研究方向为可信计算、分布式计算、信息安全等。

率。

## 1 相关工作

1994 年 Marsh 首次系统地论述了信任的形式化问题<sup>[18]</sup>,为信任机制应用到计算机系统中奠定了基础,此后各种应用环境中的信任生成算法得到了广泛研究。

### 1.1 Beth 信任生成算法

Beth 信任生成算法<sup>[5]</sup>的信任值由直接信任和推荐信任两部分综合得到,设  $P_i$  是推荐路径上不同的推荐实体,  $V_{i,j}$  表示各条推荐路径上最终推荐信任值,  $n_i$  表示  $P_i$  作为最终推荐实体所拥有的关系数,用公式 (1) 计算其直接信任值。

$$V_{\text{com}} = 1 - \prod_{i=1}^m n_i \sqrt{\prod_{j=1}^{n_i} 1 - V_{i,j}} \quad (1)$$

该计算方法考虑了同一个推荐者出现在不同推荐路径上的情况。针对同一经验在不同路径上有可能产生不同的推荐结果,采用取算术平均值的方法得到唯一信任值。

### 1.2 EigenTrust 信任生成算法

在 EigenTrust 信任生成算法<sup>[2,3]</sup>中当用户  $i$  想了解用户  $k$  的信任信息时,首先从曾与  $k$  发生过交互的用户处获得用户  $k$  的信任值,然后  $i$  根据自己对这些用户的信任情况用公式 (2) 计算得到关于用户  $k$  的信任值  $T_k$ 。

$$T_k = \sum_j (C_{ij} \times C_{jk}) \quad (2)$$

$C_{ij}$  为任意用户  $i$  对  $j$  的直接信任值。 $T_k$  为用户  $k$  的全局信任值。算法中还给出了信任形式化推导的具体规则。

### 1.3 PeerTrust 信任生成算法

PeerTrust 信任生成算法<sup>[4]</sup>在计算信任值时引入了满意度反馈、交互数量、交互环境等五个因素,信任值相对较为准确,其信任值计算方法如公式 (3) 所示。

$$T(u) = \alpha * \frac{\sum_{i=1}^{I(u)} S(u, i) \times C_r(p(u, i)) \times TF(u, i)}{I(u)} + \beta * CF(u) \quad (3)$$

$S(u, i)$  表示第  $i$  次交互的满意程度,  $C_r(p(u, i))$  表示反馈可信程度,  $TF(u, i)$  表示本次交互环境,  $CF(u)$  表示给定时期内节点  $U$  所处的社区环境,  $\alpha, \beta$  表示两部分在综合值  $T(u)$  中的权重。

## 2 基于反馈的信任生成算法 SFAF

由于安全多方计算环境的特殊性,可以认为一次合作计算的结果只有成功和失败两种情况。算法 SFAF 信任值的变化规律采用“慢增长,快下降”,信任

值的提高是缓慢的过程,要经过多次成功合作计算才能达到一个较高的信任值,但只要有一次计算失败信任值就迅速下降,且不诚实计算次数越多信任值下降比例越大。经过多次不诚实计算恶意参与者的信任值很低,从而可隔离恶意参与者,降低计算复杂度,提高计算效率。

### 2.1 算法 SFAF 描述

为实现基于反馈的信任值按“慢增长,快下降”的规律变化,将信任值  $T(n)$  表示成连续成功合作次数 ( $k$ ),合作诚实情况 ( $\delta$ ),合作失败次数 (fail) 和最近一次更新完成时的信任值  $T(n-1)$  的函数。当参与者初次进入系统时,系统中其他参与者并不了解该参与者的诚实情况处于半信半疑状态,将其初始信任值  $T(0)$  设置为 0.5。

每次有参与者合作计算完成,如果是成功的则用公式 (4) 更新信任值  $T(n)$ ,同时连续成功合作次数  $k$  增加 1。

$$T(n) = T(n-1)^{k^{-1/\alpha}} \quad (4)$$

其中  $\alpha$  是根据环境需要设置的常数,若是环境中恶意参与者较多可以设置  $\alpha$  值较大使得信任值增长更加缓慢。 $k$  是连续成功合作次数,由公式 (4) 的数学特性可知参与者信任值  $T(n)$  随着连续成功合作次数  $k$  增加而缓慢增大。有合作计算完成,如果是失败的则用公式 (5) 更新信任值  $T(n)$ ,同时连续成功合作次数  $k$  置 0,合作失败次数 fail 增加 1。

由公式 (5) 的数学特性可知一旦发生合作失败信任值就会迅速衰减。

$$T(n) = T(n-1) * \delta \quad (5)$$

衰减系数  $\delta$  可根据不同的应用环境需要设置不同的初值。如果系统中常有恶意参与者用诚实行为骗取高信任值,可以设置较小的  $\delta$  值,甚至为 0,这样可有效抵制参与者的信任欺骗。当发生合作计算失败时信任值衰减系数用公式 (6) 衰减变化。

$$\delta = \delta / \text{fail} \quad (6)$$

其中 fail 是失败合作的次数。恶意参与者的信任值衰减系数  $\delta$  随着失败合作次数 fail 的增加不断下降,从而信任值衰减幅度增大,每次信任值衰减都需要更多次的成功合作才能得到恢复;当失败合作次数 fail 很大时信任值衰减系数  $\delta$  将接近 0,信任值  $T(n)$  也会接近 0,从而可彻底隔离恶意参与者。

### 2.2 算法代码描述

根据上述算法思想,给出 C 语言伪代码描述算法 SFAF 如下:

SFAF (slow grow and fast down value generating algorithm)

```

{
    fun = Function (); /* 合作成功 Function() 返回
1, 否则返回 -1 */
    if (fun == 1) /* 合作成功 */
    { 计算更新直接信任值;
      fun = 0; /* 重置 fun 值 */
      k = k + 1; /* 成功计数器加一 */
    }
    if (fun == -1) /* 合作失败 */
    { 计算更新直接信任值;
      fun = 0;
      fail = fail + 1; /* 失败计数器加一 */
      k = 0; /* 成功次数置 0 */
       $\delta = \delta / \text{fail}$ ; /* 减小衰减系数 */
    }
}

```

### 2.3 算法复杂度分析

算法要求每个参与者维护一张信任表,表中保存所有有过交互记录参与者的信任值、信任值计算时间、连续成功合作计算次数和合作失败次数。参与者维护的表结构如表(1)。随时间和环境的变化,参与者的可信情况也会发生变化,根据环境预先设定一个信任值有效时间  $\Delta T$ 。每次有参与者请求合作计算时,首先查询信任表,如果表中有其信任值,且表中信任值在有效时间  $\Delta T$  范围内,则根据信任值采用相应的安全策略,如果表中没有信任值或表中的信任值已经超过有效时间  $\Delta T$ ,则重置参与者信任值为初始值。

表 1 参与者信任表

参与者名称	信任值	计算更新时间	成功次数 ( $k$ )	失败次数 ( $\delta$ )
$P_1$	$T_{p1}$	$\text{Time}_1$	$k_1$	$\delta_1$
$P_2$	$T_{p2}$	$\text{Time}_2$	$k_2$	$\delta_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$P_n$	$T_{pn}$	$\text{Time}_n$	$k_n$	$\delta_n$

假设环境中  $n$  位参与者,其中  $m$  位参与者合作完成一个计算。当合作计算完成时,每个参与者根据合作计算结果独自计算更新对其他参与者的信任值,更新信任表。每位参与者计算对其他  $m-1$  位参与者的信任值,信任值计算的时间复杂度为  $O(m^2)$ ,在计算中参与者之间没有通信,算法的通信复杂度为  $O(1)$ 。

## 3 仿真实验及性能分析

采用 Visual C++ 6.0 仿真文中信任生成算法,仿真实验中设置 100 个参与者。实验一中恶意参与者为 30%,实验二将实验结果与 EigenTrust 信任生成算

法做了对比。

### 3.1 仿真实验步骤

Step1 产生参与者编号  $P_i$ ,设置参与者相关参数。参与者的信任值  $T(n)$  和信任值衰减系数  $\delta$  的初值均设置为 0.5,常数  $\alpha$  的初值设为 2。

Step2 随机产生数字  $i$  作为参与者的编号。

Step3 查询信任表获得参与者  $i$  的当前信任值及上次信任值更新时间,实验中假定恶意参与者的有效信任值达到 0.5 以上时所进行的合作计算就是不成功的。

Step4 根据 Step3 的结果,统计失败计算和成功计算的次数,计算更新参与者当前信任表。

Step5 重复 Step2~Step4,分析计算合作成功率。

### 3.2 仿真实验分析

从图 1 中可以看出,随着合作计算次数增加采用算法 SFAF 控制的成功率不断上升。多次合作计算后根据算法 SFAF 的评价,恶意参与者信任值很低,根据信任值隔离恶意参与者后,合作成功率明显上升。在环境中恶意参与者比例达 30% 时,当合作计算次数到 2300 次时成功率已经超过 90%,当合作计算次数达到 1.55 万次时,由于恶意参与者已经基本完全隔离,合作成功率可高达 95% 以上,有效遏制了恶意参与者信任欺骗,达到了预期目标。从图 2 中可以看出随着恶意参与者比例增加算法 SFAF 的合作成功率明显高于 EigenTrust 信任生成算法,当恶意参与者比例高达 50% 时, EigenTrust 信任生成算法合作成功率只有 61%,而算法合作成功率仍然高达 80% 以上。

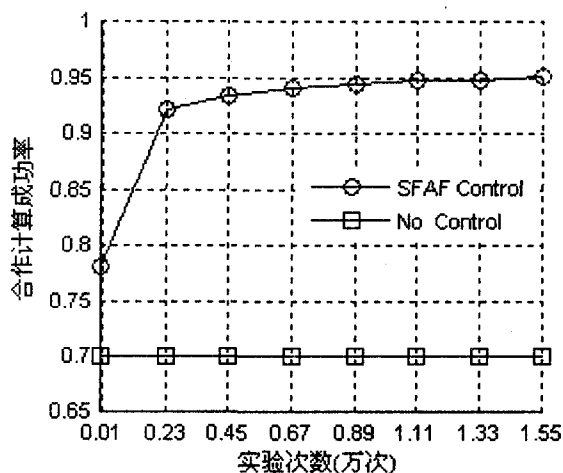


图 1 不同试验次数合作成功比率图

## 4 结束语

文中提出了一种适合安全多方计算环境的信任生成算法 SFAF,该算法中参与者的信任值更新采用慢增长快下降的方法。参与者之间经过多次合作计算

后,诚实参与者的信任值很高,而恶意参与者的信任值会维持在较低水平。根据信任值的高低可以有效隔离恶意参与者,提高合作成功率。文中仅提出了信任值生成算法,后面将进一步研究信任值的推荐传播。

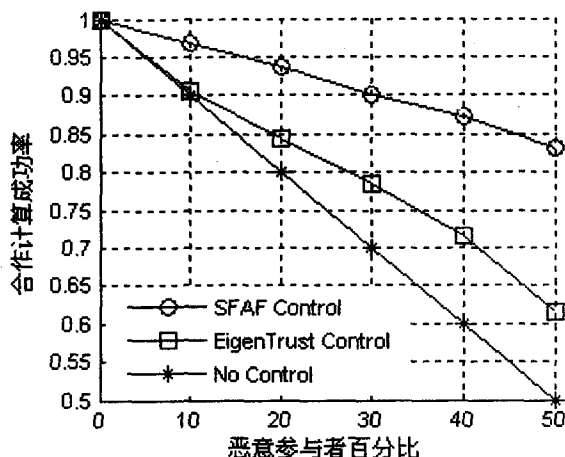


图 2 不同恶意参与者比例合作成功比率图

#### 参考文献:

- [1] Blaze M, Feigenbaum J, Lacy J. Decentralized trust management[C]//Proceedings of the 17th Symposium on Security and Privacy. Oakland: IEEE Computer Society Press, 1996: 164-173.
- [2] Kamvar S D, Schlosser M T, Garcia-Molina H. The EigenTrust Algorithm for Reputation Management in P2P Networks [C]//Proceedings of the 12th International World Wide Web Conference. Budapest, Hungary: ACM Press, 2003: 640-651.
- [3] 唐文, 陈钟. 基于模糊集合理论的主观信任管理模型研究[J]. 软件学报, 2003, 14(8): 1401-1408.
- [4] Li Xiong, Liu Ling. Peer Trust: Supporting reputation-based trust in peer-to-peer communities[J]. IEEE Transactions on Data and Knowledge Engineering, 2004, 16(7): 843-857.
- [5] Beth T, Borcherding M, Klein B. Valuation of Trust in Open Networks[C]//Proceedings of the European Symposium on Research in Computer Security (ESORICS). Brighton: Springer-Verlag, 1994: 3-18.
- [6] 田春岐, 邹仕洪, 王文东, 等. 一种基于推荐证据的有效抗攻击 P2P 网络信任模型[J]. 计算机学报, 2008, 31(2): 270-281.
- [7] Josang A. A model for trust in security systems[C]//Proceedings of the 2nd Nordic Workshop on Secure Computer Systems. New York: ACM Press, 1997.
- [8] Josang A. A logic for uncertain probabilities[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 2001, 9(3): 279-311.
- [9] 胡建理, 吴泉源, 周斌. P2P 环境下基于信誉的信任模型研究[J]. 计算机学报, 2009, 36(9): 1-6.
- [10] Yao A C. Protocols for secure computations[C]//Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS). Chicago, USA: [s. n.], 1982: 160-164.
- [11] 罗永龙, 黄刘生, 荆巍巍, 等. 一个保护隐私的布尔关联规则挖掘算法[J]. 电子学报, 2005, 33(5): 900-903.
- [12] Luo Yonglong, Huang Liusheng, Zhong Hong, et al. A secure protocol for determining whether a point is inside a convex polygon[J]. Chinese Journal of Electronics, 2006, 15(4): 578-582.
- [13] 罗永龙, 黄刘生, 荆巍巍, 等. 保护私有信息的叉积协议及其应用[J]. 计算机学报, 2007, 30(2): 248-254.
- [14] Luo Yonglong, Huang Liu sheng, Zhong Hong. Secure Two-Party Point-Circle Inclusion Problem[J]. Journal of Computer Science and Technology, 2007, 22(1): 88-91.
- [15] 罗永龙, 黄刘生, 徐维江, 等. 一个保护私有信息的多边形相交判定协议[J]. 电子学报, 2007, 35(4): 685-691.
- [16] 张彩云, 罗永龙, 石磊. 一个点与矩形区域包含关系的安全判定协议[J]. 计算机技术与发展, 2009, 19(9): 140-142.
- [17] 石磊, 罗永龙, 张彩云. 随机化算法及其在最小外接圆求解中的应用[J]. 计算机技术与发展, 2009, 19(8): 82-88.
- [18] Marsh S P. Formalizing trust as a computational concept[D]. Stirling: University of Stirling, 1994.
- [19] 张建安, 高晓光. GSPN 的分析方法及其应用[J]. 火力与指挥控制, 2005, 30(5): 65-67.
- [20] Molly M K. Discrete Time Stochastic[J]. IEEE Transactions on software Engineering, 1985, 31(4): 417-423.
- [21] 许春霞, 姜浩. 基于随机 Petri 网的工作流仿真[J]. 计算机技术与发展, 2009, 19(4): 87-90.
- [22] 张力. Petri 网模型与程序流程图的比较及应用研究[J]. 计算机技术与发展, 2006, 16(6): 151-154.
- [23] Donatelli S, Sereno M. On the Product form Solution for Stochastic Petri Nets [C]//In: Proc. of IEEE in FOCOM 2004. San Francisco, CA, USA: [s. n.], 2004.
- [24] 卢岚, 乔利. 广义随机 Petri 网在 BPR 中的应用研究[J]. 软科学, 2006, 20(1): 45-48.
- [25] 罗雪山. Petri 网在 C4ISR 系统建模、仿真与分析中的应用[M]. 长沙: 国防科技大学出版社, 2007.
- [26] Molly M K. Performance Analysis Using Stochastic Petri Nets [J]. IEEE Transactions on Computer, 1982, 31(9): 913-917.
- [27] 孟海宁, 齐勇, 侯迪. 基于非马尔科夫随机 Petri 网的软件再生建模与分析[J]. 计算机学报, 2007, 30(12): 2212-2215.

(上接第 161 页)