动态异构冗余结构的拟态防御自动机模型

朱维军[1,2],郭渊博[3],黄伯虎[4]
(1. 郑州大学信息工程学院,河南郑州 450001;2. 北京大学信息科学技术学院,北京 100871;
3. 信息工程大学密码工程学院,河南郑州 450001;4. 西安电子科技大学计算机学院,陕西西安 710071)

摘 要: 动态异构冗余结构是拟态防御技术的常用工程模型.然而,目前尚缺乏对该结构实施形式化分析的手段,因为该结构缺乏形式化建模方法.针对此问题,使用有穷状态自动机及其并行组合自动机为一些拟态攻防行为建立计算模型.首先,使用单个有穷状态自动机为单个执行体建模;其次,使用有穷状态自动机的并行组合为执行体组合建模;再次,修改状态迁移规则,得到可描述攻防行为的拟态防御自动机模型;最后,根据该自动机模型的状态条件,分析动态异构冗余结构上拟态攻防行为的安全性.此外,也可使用交替自动机为拟态攻防建模,并把安全性自动分析规约为交替自动机模型检测问题.

关键词: 动态异构冗余;拟态防御;自动机
中图分类号: TP301.1、TP309 **文献标识码:** A **文章编号:** 0372-2112 (2019)10-2025-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.10.002

A Mimic Defense Automaton Model of Dynamic Heterogeneous Redundancy Structures

ZHU Wei-jun[1,2], GUO Yuan-bo[3], HUANG Bo-hu[4]
(1. *School of Information Engineering, Zhengzhou University, Zhengzhou, Henan* 450001, *China*;
2. *School of Electronics Engineering and Computer Science, Peking University, Beijing* 100871, *China*;
3. *Cryptography Engineering Institute, Information Engineering University, Zhengzhou, Henan* 450001, *China*;
4. *School of Computer Science, Xidian University, Xi'an, Shaanxi* 710071, *China*)

Abstract: Up to now, the Dynamic Heterogeneous Redundancy (DHR) structure is a kind of important engineering model about the Mimic Defense (MD) technique. However, there is still a lack of way of formal analysis for DHR structures as there is no formal model available for a DHR structure. To address this problem, we use a Finite State Automaton (FSA) and its Parallel Automaton (PA) to establish a computing model for some attacks and mimic defenses. First, each FSA is employed to model each execution body, while there are a number of execution bodies in a DHR structure. Second, these FSAs are combined in parallel to model the combination of execution bodies. Third, one can get a model of MD automaton which can describe the attacks and MD actions, by modifying the state transition rules. Finally, one can analyse the attacks and MD actions on a DHR structure, according to the conditions of the PA states. Furthermore, we use an Alternating Finite Automaton (AFA) to model some attacks and MD actions. As a result, the automatic MD analysis problem is reduced to the solved AFA model checking problem.

Key words: dynamic heterogeneous redundancy; mimic defense; automata

1 引言

网络安全事关国家安全.然而,"网络空间的基本安全态势是易守难攻"[1],这是因为从本质上来说,被动防御技术一定程度上造成了当前网络空间安全的被动态势.为此,在邬江兴院士的领军下,拟态防御(Mimic Defenses, MD)技术被提出[2].从已有的文献报道来看,动态异构冗余(Dynamic Heterogeneous Redundancy, DHR)结构是 MD 技术的一个主流工程模型[1,3,4].仿真、实验与测试已证明 DHR 拟态防御可大幅提升系统

收稿日期:2018-09-21;修回日期:2019-07-26;责任编辑:李勇锋
基金项目:国家自然科学基金(No. U1204608);国家重点研发计划(No. 2016YFB0800100)

的安全性[1,3,4]. 例如:在 2019 最新一次国际大赛中,拟态防御成功顶住了来自中、美、俄等国 29 支国际顶尖白帽黑客战队发起的 290 万次攻击[5].

在非拟态的经典计算中,形式化建模基础上的自动分析技术可在状态空间上搜索不安全状态,可弥补抽样测试方式的不足. 这样的形式化分析技术之所以通用、有效,是因为:(1)使用的形式化模型足够抽象,因而可满足通用性;(2)使用的形式化模型不能过于抽象,它适度具体到可描述系统运行的过程,从而满足有效性.

对于拟态版的防御系统而言,情况类似. 为了对拟态防御系统实施更加自动化的分析,需要适度抽象的形式化模型. 一方面,DHR 模型位于工程层面,需要一整套形式化符号系统在抽象层面提供支撑,以便利用形式化的分析工具提供自动化分析. 另一方面,拟态安全系统[6]与文献[7]中的模型均为形式化模型,它们非常有用,但不适合用于以自动化分析为目的的建模,因为抽象程度过高,难以刻画系统运行的内在行为过程,无法对这些行为实施分析,只适合描述性质.

因此,就 DHR 拟态防御来说,还需要一种形式化、自动化的建模与分析技术,以实现可全面验证拟态系统安全性. 这是本工作研究的课题.

自动机是非拟态系统形式化分析的常用形式化建模模型,同理,如果我们可以使用某种自动机为拟态防御建模,就可以利用自动机验证工具分析拟态防御系统的行为. 遗憾的是,已有的自动机无法直接为拟态防御建模,因为不能直接刻画拟态行为,因此需要给出新的自动机模型. 此外,如果这种新自动机可以通过变换规约为已有的自动机模型,则调用已有的自动机验证工具可用于分析拟态防御系统的行为. 这就是本项工作的研究思路.

2 相关工作

2.1 拟态的形式化理论与模型

拟态防御的形式理论被提上了日程[8]. 拟态防御系统被描述为一个三元组[6]. 控制论、博弈论、稀疏表达理论被设想用于分析拟态安全系统[6]. 顺理成章地,拟态核心思想——拟态变换被认为可以由随着时间而变化的状态变化来表征[8],这里蕴含状态迁移的思想,但缺乏具体状态迁移过程描述,后者正是本工作的一个特点.

概率与统计在拟态防御理论研究中始终扮演了重要角色. DHR 结构的抗攻击能力如何? 一次攻击的成功概率公式、对特定漏洞的攻击成功率公式均被证明[1]. 此外,攻击成功的概率密度函数、防御性能与多种相关因素之间的关系被探明[4]. 这些工作从数学层面探明拟态防御的强大效果与效率规律、使用范围,指导工程开发设计,因此非常重要,但并不对具体运行过程建模,这是与本研究的一个区别.

2.2 拟态防御的架构、算法及其应用

拟态防御系统设计的一个关键环节是异构功能体调度[9],给出更多的调度算法可为不同需求的用户提供灵活的方案. 此外,拟态防御的动态、异构设计思想也被用于多个领域[10~15]. 近年来这方面的工作较多,篇幅所限,不再详述.

3 基础知识

3.1 DHR 结构

动态异构冗余结构的思想与原理,参见文献[1].

使用异构冗余(非相似余度)构造,根据“相对正确”原理[16],多数执行体未被攻破并正确安全地实施计算是一个大概率事件. 因此,表决器可认为多数执行体计算的结果是可采纳的. 然而,非相似余度构造的计算结构是静态不可变的. 故此动态思想被引入,被选择参与计算的执行体集合仅仅是所有构件集合的一个子集,每个构件都是一个可独立执行计算的单元,动态选择算法随机调整参与计算的执行体集合. 表决器与反馈机制就决定了,动态异构冗余机制下的拟态防御比普通防御技术具有更好的安全性,详情参见文献[16].

3.2 自动机

定义 1 有穷状态自动机(Finite State Automata, FSA)是一个五元组 $M=(Q,\Sigma,\delta,q_0,F)$,其中:

Q 是有穷状态集,$\forall q\in Q$,q 称为一个状态.

Σ 是输入字母表.

δ 是状态迁移函数,$\delta:Q\times\Sigma\to Q$,$\delta(q,a)=p$.

q_0 是 M 的初始状态,$q_0\in Q$.

F 是 M 的终止状态集合,F 被 Q 包含. 任给 $q\in F$,q 称为 M 的终止状态.

有了有穷状态自动机的定义,就可在此基础上定义 FSA 的组合/并行操作. 简单地说,由 n 个 FSA 使用“‖”操作组合而成的一个并行 FSA 是这样的:它的一个状态具有形式$(q_1^{(i)},q_2^{(i)},\cdots,q_n^{(i)})$,其中,$q_1^{(i)}$ 是第 1 个成分 FSA 运行过程的第 i 个状态;该并行 FSA 的一个输入字符具有形式$(a_1^{(i)}\wedge a_2^{(i)}\wedge\cdots\wedge a_n^{(i)})$,其中,$a_1^{(i)}$ 是第 1 个成分 FSA 的第 i 步状态迁移对应的输入字符;该并行 FSA 的一个状态迁移具形式 $\delta'((q_1^{(i)},q_2^{(i)},\cdots,q_n^{(i)}),(a_1^{(i)}\wedge a_2^{(i)}\wedge\cdots\wedge a_n^{(i)}))=(q_1^{(i+1)},q_2^{(i+1)},\cdots,q_n^{(i+1)})$;该并行 FSA 的初始状态具有形式$(q_1^{(1)},q_2^{(1)},\cdots,q_n^{(1)})$;该并行 FSA 的终止状态具有形式$(f_1,f_2,\cdots,f_n)$,其中 f_j 是第 j 个成分 FSA 的一个终止状态.

3.3 交替自动机与交替时序逻辑

交替自动机(Alternating Finite Automata, AFA)[17]可用来描述两个行为主体的交互行为,常用于博弈建

模. 交替时序逻辑(Alternating Temporal Logic, ATL)[18]可用于描述交互时序性质. 如果使用 AFA 建模,使用 ATL 表达性质,那么,存在一个模型检测算法[19]可用于自动判定 AFA 是否满足 ALT 性质.

4　拟态防御自动机

本节给出 DHR 结构的自动机模型. 我们的自动机使用有穷状态自动机及其组合/并行操作建模.

例 1　图 1 给出了一个 DHR 结构的自动机运行实例. 在该例中,有 3 个执行体参与计算,另有 2 个构件作为备用待参与计算. 在计算的第一阶段(运行的前两个状态):执行体 1、执行体 2 和执行体 3 参与计算;执行体 4(构件 4)和执行体 5(构件 5)待参与计算. 在计算的第二阶段(运行的后两个状态):执行体 1、执行体 3 和执行体 4 参与计算;执行体2(构件2)和执行体5(构件5)待参与计算. 图中:红色填充圆表示构件状态;白色填充圆表示执行体状态;蓝色填充圆表示不安全状态.

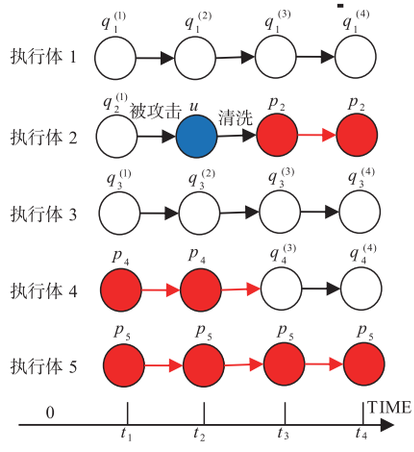


图1　DHR结构的一个自动机运行实例

在图 1 中,5 个 FSA 分别为上述 5 个执行体(构件)建立自动机模型,每个 FSA 的运行均通过 4 个状态. 该图可见,执行体 1 和执行体 3 全程参与计算,它们并未遭受攻击. 执行体 2 在运行进行到第一个状态后即遭到攻击,这导致它进入不安全状态. 根据 DHR 结构原理,当这种情况发生时,多个执行体计算结果出现不同,表决器选择多数执行体的计算结果,并且对出现错误的执行体 2 做清洗操作,同时从构件集中随机选择构件 4(执行体 4)代替执行体 2 参与计算. 上述拟态防御机制发挥作用后,参与计算的执行体数量和作为备选的构件数量仍保持不变,系统继续运行.

图 2 表达出了系统运行状态与时间流逝、发生事件(攻击、表决反馈)之间的关系. 如果说图 1 中 5 个执行体的第一个状态分别是 $q_1^{(1)}, q_2^{(1)}, q_3^{(1)}, p_4, p_5$,那么这 5 个执行体的第一个状态的组合状态是$(q_1^{(1)}, q_2^{(1)}, q_3^{(1)}, p_4, p_5)$,参见图 2. 以此类推,图 2 中可见更多的组合状态:$(q_1^{(2)}, u, q_3^{(2)}, p_4, p_5)$ $(q_1^{(3)}, p_2, q_3^{(3)}, q_4^{(3)}, p_5)$ $(q_1^{(4)}, p_2, q_3^{(4)}, q_4^{(4)}, p_5)$. 上述 4 个组合状态构成了新自动机的状态集. 显然,新自动机的状态迁移关系是 5 个成分 FSA 的状态迁移的并操作,新自动机状态迁移的输入字符是 5 个成分 FSA 相应状态迁移的输入字符的并.

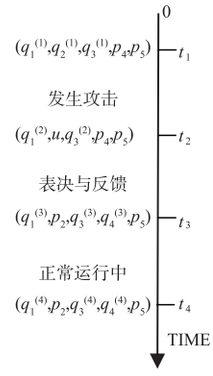


图2　多执行体组合状态(并行自动机运行)

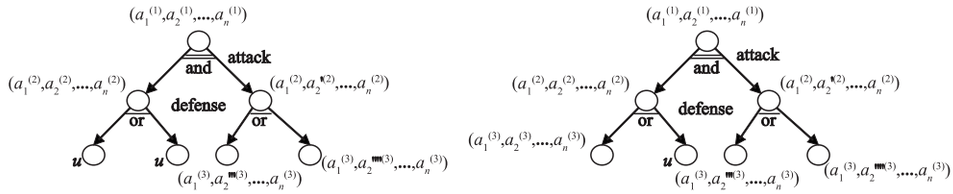
例 1 可见,可通过多个 FSA 的并操作(‖)获得的并行自动机为 DHR 拟态防御建立自动机模型.

定义 2　若 FSA $A_1, A_2, \cdots, A_n$ 分别为 n 个执行体(构件)建模,则 $A = A_1 \parallel A_2 \parallel \cdots \parallel A_n$ 是由这 n 个执行体组成的 DHR 结构的拟态防御自动机模型.

定义 2 事实上给出了未被攻击的 DHR 结构的自动机 A 的建模算法. 若 DHR 结构遭到攻击,用于描述攻击与防御行为的并行自动机行为模型 A'可通过算法 1 获得.

算法 1　构造发生攻击时的拟态防御自动机模型

INPUT: $\forall i \in n$,执行体 i 的自动机模型 A_i
OUTPUT:发生攻击时的拟态防御自动机模型 A'
BEGIN
$A := A_1 || A_2 || \cdots || A_n$
$A' := A$
If $\exists u$ 使得 A'中存在如下形式的状态迁移规则 $\delta((q_1^{(i)}, \cdots, q_{m-1}^{(i)}, q_m^{(i)}, q_{m+1}^{(i)}, \cdots, q_n^{(i)}, p_{n+1}, \cdots, p_o), attack)$
$= (q_1^{(i+1)}, \cdots, q_{m-1}^{(i+1)}, u, q_{m+1}^{(i+1)}, \cdots, q_n^{(i+1)}, p_{n+1}, \cdots, p_o)$
(迁移规则 1)
If $|u| < n/2$ then / * $|u|$表示不安全状态 u 在迁移规则 1 中出现的次数
对于所有的 u,把迁移规则 1 替换为
$\delta'((q_1^{(i)}, \cdots, q_{m-1}^{(i)}, q_m^{(i)}, q_{m+1}^{(i)}, \cdots, q_n^{(i)}, p_{n+1}, \cdots, p_o), \tau)$
$= (q_1^{(i+1)}, \cdots, q_{m-1}^{(i+1)}, p_m, q_{m+1}^{(i+1)}, \cdots, q_n^{(i+1)}, q_{n+1}^{(i+1)}, p_{n+2}, \cdots, p_o$
$\overset{m与n+1序号互换}{=} (q_1^{(i+1)}, \cdots, q_{m-1}^{(i+1)}, q_m^{(i+1)}, q_{m+1}^{(i+1)}, \cdots, q_n^{(i+1)}, p_{n+1}, p_{n+2}, \cdots, p_o)$
Else if $|u| \geq n/2$ then 把迁移规则 1 替换为
$\delta'((q_1^{(i)}, \cdots, q_{m-1}^{(i)}, q_m^{(i)}, q_{m+1}^{(i)}, \cdots, q_n^{(i)}, p_{n+1}, \cdots, p_o), \tau) = u$
End if
End if
END



(a) 不满足弱安全　(b) 满足弱安全，不满足强安全

(c) 满足弱安全，不满足强安全　(d) 满足强安全

图3　DHR结构的博弈树

已使用 cpu 时间生成 09-Jun-2019 19:56:29。

函数名称	调用	总时间	自用时间*
strong_security_200000	1	3.259 s	1.000 s
xlsread	1	2.259 s	0.110 s

(a) 20万个节点

已使用 cpu 时间生成 09-Jun-2019 19:58:13。

函数名称	调用	总时间	自用时间*
strong_security_400000	1	6.744 s	2.326 s
xlsread	1	4.417 s	0.219 s

(b) 40万个节点

已使用 cpu 时间生成 09-Jun-2019 20:00:00。

函数名称	调用	总时间	自用时间*
strong_security_600000	1	10.030 s	3.448 s
xlsread	1	6.583 s	0.281 s

(c) 60万个节点

已使用 cpu 时间生成 09-Jun-2019 20:01:49。

函数名称	调用	总时间	自用时间*
strong_security_800000	1	13.393 s	4.267 s
xlsread	1	9.126 s	0.406 s

(d) 80万个节点

图4　运行时间：判定强安全

已使用 cpu 时间生成 09-Jun-2019 13:59:33。

函数名称	调用	总时间	自用时间*
weak_security_20000	1	49.820 s	49.103 s
xlsread	1	0.717 s	0.015 s

(a) 2万个节点

已使用 cpu 时间生成 09-Jun-2019 14:07:58。

函数名称	调用	总时间	自用时间*
weak_security_40000	1	196.579 s	195.721 s
xlsread	1	0.858 s	0.031 s

(a) 4万个节点

已使用 cpu 时间生成 09-Jun-2019 14:24:41。

函数名称	调用	总时间	自用时间*
weak_security_60000	1	441.647 s	440.617 s
xlsread	1	1.030 s	0.032 s

(a) 6万个节点

已使用 cpu 时间生成 09-Jun-2019 14:50:10。

函数名称	调用	总时间	自用时间*
weak_security_80000	1	784.525 s	783.324 s
xlsread	1	1.201 s	0.046 s

(a) 8万个节点

图5　运行时间：判定弱安全

定义 3　DHR 结构强安全性被定义为：拟态防御自动机 A' 不存在不安全状态 u（无条件安全）.

算法 2　强安全性判定算法

```
INPUT:A′
OUTPUT:A′是否强安全(yes or no)
    BEGIN
      If A′存在状态 u then
        return“no”
      Else
        return“yes”
      End if
    END
```

算法 1 为遭受攻击的 DHR 结构建立了拟态防御自动机模型，算法 2 在该模型上判定 DHR 结构的强安全性. 事实上，即使 DHR 结构不满足强安全性（即自动机中存在不安全状态），系统仍然有可能是安全的（弱安全）. 弱安全性在交替自动机和交替时序逻辑上定义.

非形式化地讲，弱安全性被定义为：“无论攻方如

何策略,守方可采取正确的应对以确保 DHR 结构不会进入不安全状态 u".

例 2　图 3 给出了一个使用 AFA 描述 DHR 计算的例子. 用于拟态防御的 AFA 中的状态与并行自动机相同,均为组合状态. 不同在于前者的部分状态迁移存在"and"关系. 该图中,从根节点到其子节点的有向边表示攻击策略,从叶节点的父节点到叶节点的有向边表示防御策略. 图(a)中,若攻方选择向左的边,则无论守方如何选择策略,均进入不安全状态 u,因此该子图不满足弱安全性. 图(b)和(c)中,无论攻击者怎样选择边,防御者均可选择恰当的应对以避免进入不安全状态 u,因此这两个子图是弱安全的. 由于图(b)和(c)都存在不安全状态,故两子图都不满足强安全.

弱安全性可被形式定义为一个模型检测公式.

定义 4　ATL 公式 $g \triangleq <<\exists>>\neg \Diamond u$ 定义 DHR 结构弱安全性.

设 AFA 自动机 A''是 DHR 结构攻防博弈计算模型,根据定义 4 以及模型检测基本原理,弱安全性是否被满足可被规约为模型检测如下公式:$A''| = g$,即:弱安全性被满足当且仅当上述模型检测结果为"yes".

算法 3　弱安全性判定算法

```
INPUT:A''
OUTPUT:A''是否弱安全(yes or no)
    BEGIN
    调用 ATL 模型检测算法[19] 对如下公式实施检测:A''| = g
      If 上一步骤模型检测结果是"no" then
        return"no"
      Else
        return"yes"
      End if
    END
```

5　复杂度分析

算法 1 主要执行两种操作:自动机并行操作与状态迁移规则替换. 前者操作可在多项式时间完成;单次替换需要时间 $O(1)$,替换操作在双重循环之内,被执行了多项式次数,因此后者操作也在多项式时间内完成. 故此,算法 1 为多项式时间算法.

算法 2 对于自动机 A'的所有状态逐一考察,这需要线性时间. If 语句需时 $O(1)$. 故此,算法 2 为线性时间算法.

算法 3 的关键在于判定一个 AFA 模型是否满足特定的: 公式 g. 新算法是基于 turn-based 的博弈结构,这样的 AFA-ATL 模型检测具有多项式时间复杂度[18],因此算法 3 具有多项式时间复杂度.

6　仿真实验

6.1　实验目的

验证新算法的有效性与时间效率.

6.2　实验环境

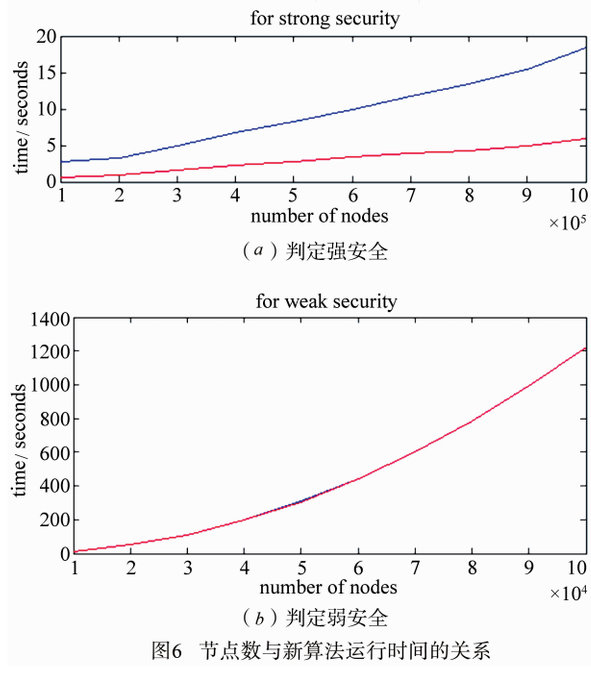
CPU:Intel core i7-4770,3.4GHz;内存:64GB;OS:windows 7 64 位;MATLAB:用于对算法 2 和算法 3 实施仿真;EXCEL:用于存储自动机信息.

6.3　实验步骤

(1)在 MATLAB 上编程实现算法 2 和算法 3;

(2)随机生成 500 个 DHR 实例,并建模获得其自动机,使用 EXCEL 文件存储每个自动机信息;

(3)执行步骤(1)所述的仿真程序,使用 MATLAB 相关函数对步骤(2)建立的 EXCEL 文件进行读写(自动机信息),并记录结果.



(a) 判定强安全

(b) 判定弱安全

图6　节点数与新算法运行时间的关系

6.4　实验结果

首先,我们获得了如下运行结果:对所有随机样本,算法判定的安全性与人工判断均一致. 这提示算法的有效性与准确性.

其次,对于强安全判定,我们获得如下结果.

图 4 给出了当自动机节点数分别为 20 万、40 万、60 万和 80 万时,仿真程序的一次运行时间,其中,"自用时间"对应于直接实现算法 2 的代码其本身的运行时间,"总时间"则是包括 MATLAB 与 EXCEL 读写交互在内的整个程序的运行时间. 该图可见:无论算法实现还是整个仿真程序,运行时间均基本呈现出线性增长. 造成这一现象的原因在于:(1)算法 2 具有线性时间复

杂度;(2)EXCEL 文件大小与自动机尺寸呈正比.

运行仿真程序 500 次,结果见图 6(a),图中蓝线涉及"总时间",红线涉及"自用时间". 该图可见以下现象:(1)无论算法实现还是整个仿真程序,运行时间均基本呈现线性增长;(2)整个仿真程序呈现出比算法实现更快增长. 前者现象与我们从图 4 中观测到的现象相同(原因也相同). 造成后者现象的原因在于:EXCEL 读写需要比算法 2 核心功能实现更长的时间,这反衬算法 2 的高效.

图 6(a)可见,自动机节点数达 10^5 量级时,算法在 10 秒之内即可完成任务,仿真程序的运行时间也不超过 20 秒;与之对照,一个高级程序员与系统分析师在同样的时间最多只能完成 200 个节点分析;自动化的新方法是人工效率的 5000 倍.

再次,对于弱安全判定,我们获得如下结果.

图 5 给出了当节点数分别为 2 万、4 万、6 万和 8 万时,仿真程序的一次运行时间. 该图可见:无论时算法实现还是整个仿真程序,运行时间均基本呈现出多项式增长. 造成这一现象的原因在于:(1)算法 3 具有多项式时间复杂度;(2)EXCEL 文件大小与自动机尺寸呈正比.

运行仿真程序 500 次,结果参见图 6(b). 该图可见以下两点现象:(1)无论时算法实现还是整个仿真程序,运行时间均基本呈现出多项式增长;(2)整个仿真程序呈现出与算法实现同样快的增长(红蓝曲线基本重合). 前者现象再一次与我们从图 5 中观测到的现象相同(原因也相同). 造成后者现象的原因在于:在算法 3 的多项式增长面前,EXCEL 的线性时间读写几乎可以忽略不计.

此外,图 6(b)可见,当节点数达 10^4 量级时,算法在 20 分钟之内可完成任务,仿真程序的运行时间也是如此;与之对照,一个高级程序员与系统分析师靠人工根本无法完成如此规模的分析.

6.5 讨论与比较

弱安全性算法核心在于调用模型检测算法. 上文实验中我们对该算法的编程实现,效率如何? 文献[20,21]开发了一个 ATL 模型检测工具,他们的 turn-based 博弈,26853 个自动机状态可在 10 秒左右完成模型检测[21],而本文实验则需要 90 秒左右才能完成此模型检测任务. 考虑到文献[21]中的 ATL 公式略微比本文算法 3 中 ATL 公式复杂,并且该文实验平台也比本文实验使用更慢的 CPU 与更小的内存,因此在同样条件下,该文模型检测代码运行速度比本文快不止 9 倍. 换句话说,我们实验中的模型检测算法的实现代码远非最优.

然而,该现象恰恰进一步反证了新方法的优点——在模型检测算法的代码实现如此"不优"的前提下,新方法尚且"表现出"如 6.4 节实验结果所示的比较优势,那么,若在工程实现层面进一步优化本文实验模型检测代码到文献[21]的水平,则新的弱安全性判定方法相对于其它潜在同类方法的比较优势又将展现如何? 答案显而易见.

7 结论

本工作在并行自动机与交替自动机的基础上定义拟态防御自动机,为 DHR 结构建立形式化模型,并给出使用新模型对 DHR 结构实施形式化分析的自动化方法. 特别是,算法 2 和算法 3 可自动分析 DHR 结构的安全性. 与测试方法相比,新方法可在状态空间实施全面高效的搜索,从而更加全面地验证 DHR 拟态防御系统. 从已有的文献看,未见可对 DHR 结构全状态空间实施自动化分析的其它方法. 这样的搜索意义在于,既不是在概率层面探讨系统安全的可能性,也不是用顶级黑客的个性化攻击手段测试安全,而是在任何攻击者都无法超越的全状态空间中全面搜索以证明安全性. 这样的安全性结论具有很强的说服力. 此外,新方法的思想也可推广到其它拟态防御乃至广义拟态计算系统. 这些均为使用新方法的益处.

参考文献

[1] 邬江兴. 网络空间拟态防御研究[J]. 信息安全学报,2016,1(4):1-10.
WU J. Research on cyber mimic defense[J]. Journal of Cyber Security,2016,1(4):1-10. (in Chinese)

[2] Hu H,Wu J,Wang Z,et al. Mimic defense:a designed-in cyber security defense framework[J]. IET Information Security,2018,12(3):226-237.

[3] 马海龙,伊鹏,江逸茗,等. 基于动态异构冗余机制的路由器拟态防御体系结构[J]. 信息安全学报,2017,2(1):29-42.
MAH,YI P,JIANG Y,et al. Dynamic heterogeneous redundancy based router architecture with mimic defenses[J]. Journal of Cyber Security,2017,2(1):29-42. (in Chinese)

[4] 扈红超,陈福才,王禎鹏. 拟态防御 DHR 模型若干问题探讨和性能评估[J]. 信息安全学报,2016,1(4):40-51.
HU H,CHEN F,WANG Z. Performance evaluations on DHR for cyberspace mimic defense[J]. Journal of Cyber Security,2016,1(4):40-51. (in Chinese)

[5] 张晔. 我拟态防御网络挡住"白帽黑客"290 万次攻击[N]. 科技日报,2019-05-24.

[6] 斯雪明. 拟态安全理论研究[EB/OL]. http://wenku.baidu.com/link?url=nZlsR3REwg_zT8K6svYQy0j14D

IUNvpxsux94ns2-ABnh7jnngc-Xb1K3c7QAPD3YmTM94xlf_53dBrjSQrLi6clZZTtp35iMpokhBV1oEW,2014-11-02.

[7] Ma Bolin, Zheng Zhang, Zhu Yongsheng. A formalization research on web server and scheduling strategy for heterogeneity[A]. Proceedings of 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference[C]. Xi'an: IEEE press, 2016. 1447 – 1451.

[8] 斯雪明,王伟,曾俊杰,等. 拟态防御基础理论研究综述[J]. 中国工程科学,2016,18(6):62 – 68.
SI X, WANG W, ZENG J, et al. A review of the basic theory of mimic defense[J]. Strategic Study of Chinese Academy of Engineering, 2016, 18(6): 62 – 68. (in Chinese)

[9] 刘勤让,林森杰,顾泽宇. 面向拟态安全防御的异构功能等价体调度算法[J]. 通信学报,2018,39(07):188 – 198.
LIU Q, LIN S, GU Z. Heterogeneous redundancies scheduling algorithm for mimic security defense[J]. Journal on Communications, 2018, 39(07): 188 – 198. (in Chinese)

[10] 仝青,张铮,张为华,邬江兴. 拟态防御 Web 服务器设计与实现[J]. 软件学报,2017,28(04):883 – 897.
TONG Q, ZHANG Z, ZHANG H, WU J. Design and implementation of mimic defense web server[J]. Journal of Software, 2017, 28(04): 883 – 897. (in Chinese)

[11] Ya-wen WANG, Jiang-xing WU, Yun-fei GUO, Hong-chao HU, Wen-yan LIU, Guo-zhen CHENG. Scientific workflow execution system based on mimic defense in the cloud environment[J]. Frontiers of Information Technology & Electronic Engineering, 2018, 19(12): 1522 – 1537.

[12] 王禛鹏,扈红超,程国振. 一种基于拟态安全防御的 DNS 框架设计[J]. 电子学报,2017,45(11):2705 – 2714.
WANG Z, HU H, CHENG G. A DNS architecture based on mimic security defense[J]. Acta Electronica Sinica, 2017, 45(11): 2705 – 2714. (in Chinese)

[13] 王禛鹏,扈红超,程国振. MNOS:拟态网络操作系统设计与实现[J]. 计算机研究与发展,2017,54(10):2321 – 2333.
WANG Z, HU H, CHENG G. Design and implementation of mimic network operating system[J]. Journal of Computer Research and Development, 2017, 54(10): 2321 – 2333. (in Chinese)

[14] 刘彩霞,季新生,邬江兴. 一种基于 MSISDN 虚拟化的移动通信用户数据拟态防御机制[J]. 计算机学报,2018,41(02):275 – 287.
LIU C, JI X, WU J. A mimic defense mechanism for mobile communication user data based on MSISDN virtualization[J]. Chinese Journal of Computers, 2018, 41(02): 275 – 287. (in Chinese)

[15] Ji X S, Huang K Z, Jin L, et al. Overview of 5G security technology[J]. Sci China Inf Sci, 2018, 61(8): 107 – 131.

[16] 邬江兴. 网络空间拟态防御导论[M]. 北京:科学出版社,2017.
WU J. Introduction to Cyberspace Mimic Defense[M]. Beijing: Science Press, 2017. (in Chinese)

[17] Satoru Miyano, Takeshi Hayashi. Alternating finite automata on ω-words[J]. Theoretical Computer Science, 1984, 32(3): 321 – 330.

[18] R Alur, T Henzinger, O Kupferman. Alternating-time temporal logic[J]. Journal of ACM, 2002, 49(5): 672 – 713.

[19] G Drimmelen. Satisfiability in alternating-time temporal logic[A]. IEEE Symposium on Logic in Computer Science[C]. US: IEEE press, 2003. 208 – 217.

[20] F Stoica, L F Stoica. Implementing an ATL model checker tool using relational algebra concepts[A]. 2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM)[C]. Split, Croatia: IEEE press, 2014. 361 – 366.

[21] Florin Stoica, Laura Florentina Stoica. Generating an ATL model checker using an attribute grammar[EB/OL]. arXiv: 1807. 08267, https://arxiv. org/abs/1807. 08267, 2019-02-05.

作者简介

朱维军　男,1976 年生于河南郑州. 郑州大学副教授. 研究方向为拟态防御、网络安全. E-mail:zhuweijun@ zzu. edu. cn

郭渊博　男,1975 年生于陕西周至. 信息工程大学教授、博士生导师,研究方向为拟态防御、信息安全.