

基于免疫算法的网络功能异构冗余部署方法

张青青, 汤红波, 游伟, 李英乐

(信息工程大学, 河南 郑州 450002)

摘要: 针对现有安全防御手段无法抵御网络功能虚拟化平台中众多未知的漏洞与后门的问题。运用拟态防御思想, 提出一种网络功能虚拟化的拟态防御架构, 并针对其中的异构体池构建问题设计了一种基于免疫算法的网络功能异构冗余部署方法。首先, 结合熵值法对异构体之间的异构度进行量化评估; 然后, 以实现异构体之间异构度最大为优化目标将网络功能异构冗余部署问题构建成极大极小问题; 最后, 基于免疫算法快速求解最优部署方案。仿真结果表明, 该方法可以迅速收敛到最优部署方案, 并保证异构体之间的异构度值整体分布在较高的水平, 有效增加了异构体池的多样性, 提升了攻击者的攻击难度。

关键词: 网络功能虚拟化; 拟态防御; 异构冗余; 免疫算法

中图分类号: TP393

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2021005

Network function heterogeneous redundancy deployment method based on immune algorithm

ZHANG Qingqing, TANG Hongbo, YOU Wei, LI Yingle

Information Engineering University, Zhengzhou 450002, China

Abstract: Aiming at the problem that the existing security defense methods cannot resist many unknown vulnerabilities and backdoors in the network function virtualization platform, a mimic defense architecture for network function virtualization using mimic defense ideas was proposed, a heterogeneous redundant deployment method based on an immune algorithm for the construction of heterogeneous pools was proposed. Firstly, the degree of heterogeneity between heterogeneous entities was quantitatively evaluated in combination with the entropy value method, then the network function heterogeneous redundant deployment problem was constructed into a minimax problem with the optimization goal of maximizing the degree of heterogeneity between heterogeneous entities, and finally the immune algorithm was used to quickly solve the optimal deployment solution of network functions. Simulation re-

收稿日期: 2020-07-17; 修回日期: 2020-11-08

通信作者: 汤红波, tahobo@sina.com

基金项目: 国家自然科学基金 (61941114, 61521003, 61801515)

Foundation Item: The National Natural Science Foundation of China (61941114, 61521003, 61801515)

论文引用格式: 张青青, 汤红波, 游伟, 等. 基于免疫算法的网络功能异构冗余部署方法[J]. 网络与信息安全学报, 2021, 7(1): 46-56.

ZHANG Q Q, TANG H B, YOU W, et al. Network function heterogeneous redundancy deployment method based on immune algorithm[J]. Chinese Journal of Network and Information Security, 2021, 7(1): 46-56.

sults show that the proposed method can quickly converge to an optimal deployment scheme and ensure that the overall distribution of heterogeneity between heterogeneous bodies is at a high level, effectively increasing the diversity between heterogeneous bodies and improving the attacker's difficulty.

Keywords: network function virtualization, mimic defense, heterogeneous redundancy, immune algorithm

1 引言

随着移动通信的发展,数据流量的爆炸式增长给移动通信网络的承载能力和运维成本带来严峻的挑战,同时随着物联网和垂直行业的快速发展,新的服务和应用层出不穷,客户的应用需求越来越多样化。现有的移动通信系统采用专用设备进行组网,具有网络改造升级困难、运营成本高等缺陷,无法满足未来网络在时延、成本以及速率等方面的指标要求。为提高网络灵活性和资源利用率、降低运营开销,5G技术在核心网中引入了网络功能虚拟化(NFV, network function virtualization)技术。与传统网络体系结构相比,NFV技术具有降低运营成本、灵活部署网络和优化资源配置等优点,但其开放共享的理念和云化架构设计给5G网络安全带来了新的挑战^[1],如不安全接口导致的用户数据泄露^[2]、控制面开放诱发的攻击表面增加^[3]、共用物理网卡带来的旁路监听风险^[4]、资源共享引发的侧信道攻击^[5]等。传统的网络防御手段,如防火墙^[6]、入侵检测^[7]等是依赖于已有漏洞、病毒的先验知识的被动防御手段,在应对NFV网络中大量未知的漏洞和后门时,由于不了解攻击者的攻击手段,无法起到相应的防御效果。

为进一步提高NFV网络的安全性,弥补现有“打补丁”式防御手段无法抵御未知漏洞与后门的缺陷,本文基于拟态防御(MD, mimic defense)^[8]思想提出一种网络功能虚拟化的拟态防御架构,利用网络系统环境的动态性和不确定性获得防御功能或属性,增加攻击者的探测和攻击难度。结合NFV的网络特性针对该防御架构中的异构体池构建问题,本文提出了一种基于免疫算法(IA, immune algorithm)的网络功能异构冗余部署方法。首先对组成异构体的软硬件组件进行异构度评估,利用熵值法将组件的异构度进行加权和构成异构体之间的异构度;然后以异构体之间的最小异构度值为优化目标,将网络功能的异构冗余

部署问题建模为极大极小问题;最后利用免疫算法求出目标函数最大的部署方案。仿真结果显示,该方法可以有效保证异构体之间的异构度,增加系统的多样性,提升攻击者的攻击难度。同时,该方法是一种冗余部署方法,虽然增加了系统的安全性和可靠性,但存在无法避免的部署开销增加问题,使请求接受率降低。

2 网络功能虚拟化的拟态防御架构

拟态防御的核心架构为动态异构冗余(DHR, dynamic heterogeneous redundancy)架构。如图1所示,DHR架构由构件池、异构体集合、调度器、输入代理、异构执行体集和裁决器组成。构件池中的软硬件实体通过多维动态重构技术可以构成一个包含 m 个异构体的集合 $E = (E_1, E_2, E_3, \dots, E_m)$,集合中异构体之间的功能相同但组成结构不同。调度器利用特定的调度策略从异构体集合中选择 n (n 为奇数)个作为执行体集 $A = (A_1, A_2, A_3, \dots, A_n)$,输入代理根据调度器的调度结果将输入信息复制成 n 份,分发给执行体集中的各个执行体进行处理。裁决器对 n 个执行体的处理结果进行多模裁决形成一致性输出,并将异常信息反馈给调度器,对调度策略进行优化改进。

由DHR架构可知,拟态防御是一种典型的内生安全机制,具有动态随机、异构冗余等内在属性。而NFV技术实现了网络功能和硬件实体的解耦分离,对底层的计算、存储、网络资源进行统一调度管理^[9],为资源的动态异构冗余配置提供了可能,与拟态防御的理念天然契合。因此,本文结合拟态防御思想提出了网络功能虚拟化的拟态防御架构。如图2所示,从NFV的架构^[10]出发采用软件层、虚拟化层和硬件层3层异构元素的多样性来实现异构性。将不同的软件功能、虚拟化软件以及硬件设备通过不同方式组合起来构成一个异构体,多个异构体组成异构体池。功能等价的异构体池为NFV架构带来了冗余性和

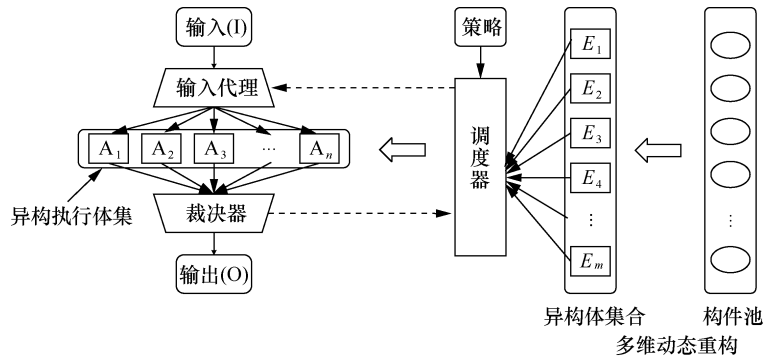


图 1 DHR 架构
Figure 1 DHR architecture

多样性, 降低了共模漏洞存在的可能性, 使攻击者无法使用相同的攻击手段对多个异构体进行连续攻击。负反馈控制器负责对异构体进行动态调度, 为 NFV 架构导入了动态性和随机性, 使系统的攻击面不断变化, 有效破坏了攻击者的持续探测行为, 阻断攻击链。多模裁决器对 n 个输出结果进行裁决, 使 NFV 架构具有了感知不明威胁的能力, 当某个执行体产生不同于其他执行体的异常输出时, 多模裁决器可以感知和定位出现异常的执行体。同时多模裁决器将单一确定目标攻击难度, 增强为多目标协同一致攻击难度, 攻击者只有成功攻击超过一半以上的异构体并产生相同错误输出才能成功攻破该防御架构, 增加了攻击者攻击成功的难度。

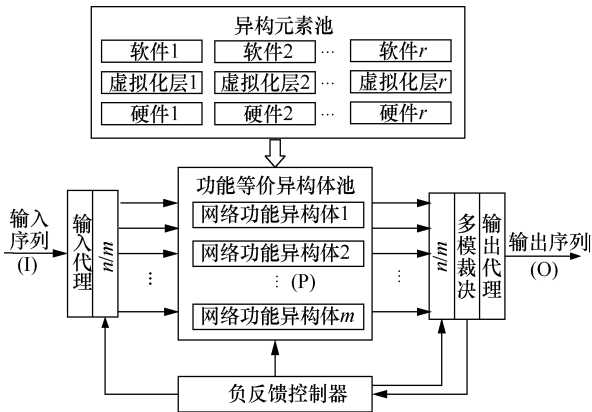


图 2 网络功能虚拟化的拟态防御架构
Figure 2 Mimic defense architecture for network function virtualization

3 问题描述与模型构建

3.1 问题描述

要实现网络功能虚拟化的拟态防御架构, 首

先要解决的问题是异构体池的构建。异构体作为处理服务请求的功能主体, 其关键特征在于异构, 异构是拟态防御架构的基础, 异构性越大, 发生共模故障的可能性越小。现有的拟态相关文献^[11-14]中, 异构体是固定的物理设备, 并且异构体的个数较少, 所以异构体池的构建相对简单。在 NFV 环境下, 虚拟化技术将软件功能与硬件平台分离, 改变了运营商的组网方式。网络虚拟化后, 传统通信网络中的网络功能以虚拟网络功能 (VNF, virtual network function) 的形式运行在物理网络中的高性能服务器上。如图 3 所示, 不同的 VNF 按照一定的逻辑顺序连接起来构成一条服务功能链 (SFC, service function chaining), 为用户提供有效的端到端服务。在 NFV 网络中, 构建异构体池就变成更为复杂的将多个功能相同的 VNF 部署到底层大量物理服务器上的问题。若不针对 NFV 网络特性设计相应的部署方法, 可能会发生异构体之间相似性较高的情况, 此时动态调度和多模裁决机制的防御效果也会大打折扣。

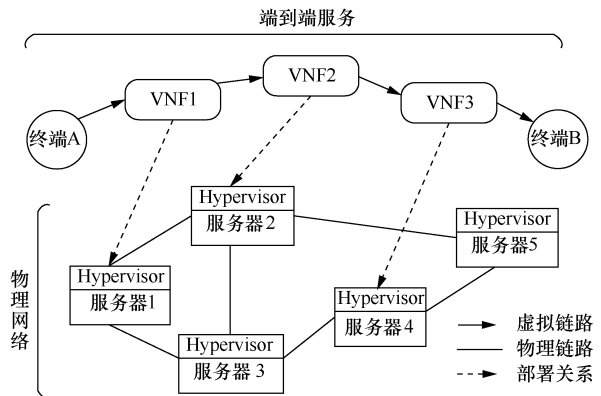


图 3 服务链请求部署
Figure 3 Service chain request deployment

3.2 系统模型

本文提出的网络功能异构冗余部署方法就是结合 NFV 网络功能软件化、资源共享化和部署动态化的网络特性,在部署 SFC 时针对每一个网络功能通过使用不同的编程语言或算法实现多个功能等价的 VNF,然后以异构度为优化目标将这些 VNF 部署到不同的服务器上得到一个功能等价的异构体池。这些异构体由 3 层异构元素组成,上层采用不同编程语言或算法实现的 VNF 是组成异构体的软件层异构元素,底层的通用服务器包含虚拟化层和物理硬件两层异构元素。虚拟化层包含不同的操作系统和虚拟化软件,操作系统可以采用 Windows、Ubuntu、Centos 等,虚拟化软件可以采用 VMware、VirtualBox、XEN 等;物理硬件层可以使用不同厂商的服务器。如图 4 所示,为 SFC 中的每一个网络功能构建包含 3 个异构体的异构体池,异构体池中的各异构体具有相同的网络功能,但其软硬件组成是不同的。

为更方便地进行描述,定义 SFC 中所有网络功能集合为 V ,各个网络功能 $v=(v_1, v_2, \dots, v_m)$,其中 v_m 表示网络功能 v 的第 m 个软件层异构元素, m 为异构体池中异构体的数量。对于每一个网络功能 $v \in V$, 其所需的计算资源记为 $\text{com}(v)$, 所需的存储资源记为 $\text{mem}(v)$, 所需的网络资源记为 $\text{net}(v)$ 。定义 S 为底层物理服务器集合,服务器 $s=(x, y)$ 由虚拟化层异构元素 x 和硬件层异构

元素 y 组成,采用 $\text{com}(s)$ 、 $\text{mem}(s)$ 、 $\text{net}(s)$ 表示服务器 $s \in S$ 所能提供的计算、存储和网络资源。

3.3 异构度量量化

异构体由各层异构元素组成,同一层的异构元素之间由于种类的不同,它们之间的差异大小也不同。各层异构元素的差异决定了异构体之间的差异性,因此可以通过对各层异构元素的异构度进行量化评估,进而得到异构体之间的异构度指标。本文借鉴了文献[15]对异构体间差异性的评价方法,并利用熵值法对其进行了改进。

针对异构元素的不同种类,采用特征向量 $T_l^i (i=1,2,\dots,m, l=1,2,3)$ 表示,其中 m 为异构体个数, l 为异构元素的层数,本文的异构体由软件层、虚拟化层和硬件层 3 层异构元素组成,分别为其编号为 1、2、3。如软件层的 VNF 由 C++、Python、Go 这 3 种编程语言实现,当第 3 个异构体的软件层 VNF 由 C++实现时,其特征向量为 $T_1^3=[1,0,0]$ 。

针对同一层异构元素不同种类之间的差异,采用异构矩阵进行量化。

$$\Phi_l = \begin{bmatrix} 0 & \phi_{12}^l & \dots & \phi_{1d}^l \\ \phi_{21}^l & 0 & \dots & \phi_{2d}^l \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{d1}^l & \phi_{d2}^l & \dots & 0 \end{bmatrix} \quad (1)$$

其中, $\phi_{ab}^l (1 \leq a \neq b \leq d, 0 \leq \phi_{ab}^l < 1, l=1,2,3)$ 表示

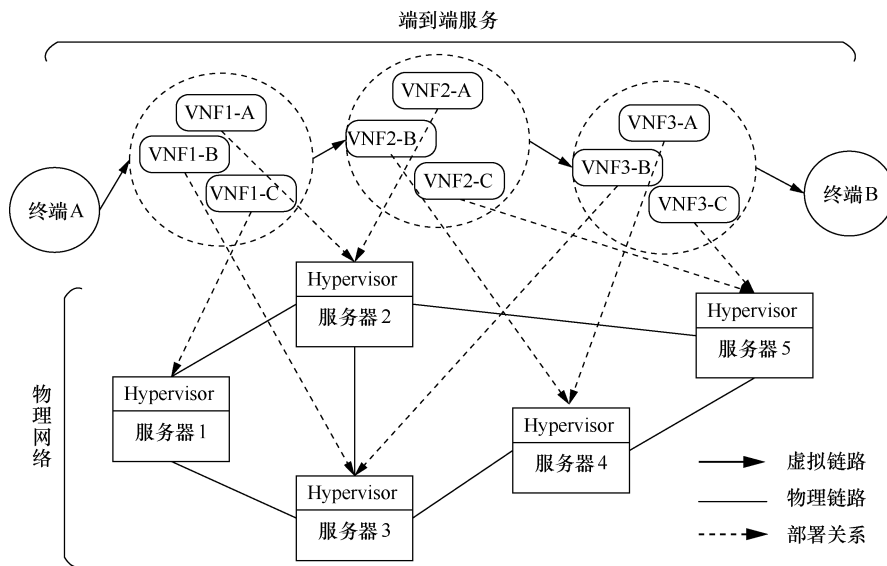


图4 网络功能异构冗余部署
Figure 4 Heterogeneous redundant deployment of network functions

l 层异构元素的第 a 种和第 b 种之间的异构度, d 是第 l 层异构元素的种类数。 ϕ_{ab}^l 的取值如式(2)所示。

$$\phi_{ab}^l = 1 - \frac{2\theta_{ab}^l}{\theta_a^l + \theta_b^l} \quad (2)$$

其中, θ_a^l 和 θ_b^l 分别表示 l 层异构元素 a 和 b 所含漏洞数量, θ_{ab}^l 表示异构元素 a 和 b 所含共模漏洞的数量, 异构元素所含漏洞信息可以通过 CVE (common vulnerabilities & exposures) 获得。 ϕ_{ab}^l 越大, a 、 b 之间的异构度越大, 当 ϕ_{ab}^l 为 1 时, 表示完全异构, 然而, 同一层异构元素的功能是等价的, 所以 a 、 b 之间总会存在一定的相似性, 无法达到理想情况; 当 $a=b$ 时, $\phi_{ab}^l=0$, 即同一种异构元素是完全相同的。异构体 i 和 j 的 l 层异构元素的异构度可由对应的特征向量和异构矩阵点乘得到, 即

$$E_l^{ij} = \mathbf{T}_l^i \boldsymbol{\Phi}_l \mathbf{T}_l^{jT} \quad (3)$$

各层异构元素的异构度加权和构成了两个异构体之间的异构度。

$$Q_{ij} = \sum_{l=1}^3 W_l E_l^{ij} \quad (4)$$

其中, W_l 是各层异构元素的权重, 具体数值采用熵值法确定。熵值法是一种客观赋权法^[16], 其基本思想是根据指标变化的大小来确定客观权重, 指标的离散程度越大, 该指标对综合评价的影响越大, 具体步骤如下。

步骤 1 考虑异构体每层异构元素有 d 种, 由异构矩阵可知每层异构元素的异构度取值有 d^2 个。将第 j 层元素的第 i 个异构度的取值记为 $\sigma_{ij} (i=1, 2, \dots, d^2, j=1, 2, 3)$ 。

步骤 2 计算第 j 层元素的第 i 个异构度值占该层的比重。由于本文统一采用 0 到 1 数值对各层元素的异构度进行量化, 并且是正向性指标, 所以此处无须进行归一化处理。

$$p_{ij} = \frac{\sigma_{ij}}{\sum_{i=1}^{d^2} \sigma_{ij}} \quad (5)$$

步骤 3 计算第 j 层元素异构度的熵值。

$$e_j = -\varphi \sum_{i=1}^{d^2} p_{ij} \ln(p_{ij}) \quad (6)$$

其中, $\varphi = \frac{1}{\ln(d^2)} > 0$ 。

步骤 4 计算各层异构度指标的权重。

$$w_j = \frac{1 - e_j}{\sum_{j=1}^3 (1 - e_j)} \quad (7)$$

3.4 数学模型构建

基于异构度指标的定义, 本节以最大化异构体之间的异构度为目标, 以底层网络中物理服务器可提供的计算资源、存储资源和网络资源为约束, 将网络功能异构冗余部署问题建模为多目标优化模型, 具体数学模型如下。

目标函数

$$\max \sum_{l=1}^3 W_l \mathbf{T}_l^i \boldsymbol{\Phi}_l \mathbf{T}_l^{jT}, 1 \leq i \neq j \leq m, \forall v \in V \quad (8)$$

约束条件

$\forall s \in S, \forall v \in V, k=1, 2, \dots, m$, 使

$$\mu_{vs}^k \text{com}(v) \leq \text{com}(s) \quad (9)$$

$$\mu_{vs}^k \text{mem}(v) \leq \text{mem}(s) \quad (10)$$

$$\mu_{vs}^k \text{net}(v) \leq \text{net}(s) \quad (11)$$

$$\sum_{k=1}^m \mu_{vs}^k \leq 1, \forall s \in S, \forall v \in V \quad (12)$$

$$\sum_{s \in S} \mu_{vs}^k = 1, \forall v \in V, k=1, 2, \dots, m \quad (13)$$

式(8)为优化目标, 表示异构体池中各异构体之间的异构度尽可能大, 式(9)、式(10)、式(11)分别表示软件层异构元素的计算、存储以及网络资源需求不能大于物理服务器中可用的计算、存储以及网络资源, 式(12)表示同一异构体池内的不同软件层异构元素不能部署在同一物理服务器上, 式(13)确保一个软件层异构元素只能部署在一个物理服务器上。其中, μ_{vs}^k 是一个 0-1 变量, 表示软件层异构元素与物理服务器之间的部署关系, 若网络功能 v 的第 k 个软件层异构元素部署到了物理服务器 s 上, 则 $\mu_{vs}^k=1$, 否则 $\mu_{vs}^k=0$ 。

由式(8)可知异构体数量为 m 时, 优化目标有

$\frac{m(m-1)}{2}$ 个。当异构体数量大于 3 时, 优化目标个数超过 3, 此时的优化问题变成高维多目标优化问题^[17], 一些经典的多目标优化算法^[18]在搜索能力、计算成本和可视化方面会遇到巨大的挑战, 优化效果大大降低。因此, 高维多目标优化问题是目前优化领域公认的研究难点。为降低求解难度, 将上述多目标优化问题转化为单目标的极大极小问题^[19], 由于攻击者大概率利用异构度最小的异构体实现协同攻击, 因此, 将异构体之间的最小异构度值作为优化目标, 可以防止出现局部极值的情况, 如式(14)所示。

$$\max \min_{1 \leq i \neq j \leq m} \sum_{l=1}^3 W_l T_l^i \Phi_l T_l^{jT}, \forall v \in V \quad (14)$$

表 1 对本节涉及的相关符号和含义进行了梳理。

4 算法描述

网络功能的部署问题已被证明是一个 NP 难问题^[20], 该问题的求解时间会随着网络规模的扩大呈指数级增长。本文将网络功能的异构冗余部署问题建模为极大极小问题。极大极小问题是一种典型的不可微优化问题^[21], 由于目标函数的梯度不连续, 现行的基于梯度概念的优化算法失效。为在尽可能短的时间内求出一个相对较优的解, 本文使用免疫算法对上述优化模型进行求解。免疫算法^[22]是受生物免疫系统启发而提出的一种智能算法, 它在遗传算法的基础上加入了免疫的概念, 在保留原算法优良特性的前提下, 利用待求问题中的一些特征信息来抑制优化过程中出现的退化现象。

(1) 产生初始抗体群

随机产生 N 个抗体组成一个抗体群, 每个抗

体都是一个异构体池的部署方案。采用实数编码方式对抗体进行编码, 抗体的长度为 m (m 为异构体的数量), 抗体中编码的取值表示相应的软件层异构元素部署的物理服务器序号, 由于底层网络有 100 个物理服务器, 所以抗体编码取值为 1 到 100 的整数。图 5 是一个抗体编码示意, 由图可知一个异构体池由 5 个异构体组成, 各异构体软件层的异构元素分别部署到物理服务器 7、23、60、5、36 上。

7	23	60	5	36
---	----	----	---	----

图 5 抗体编码示意
Figure 5 Antibody code

(2) 解的多样性评价

① 抗体与抗原间亲和力 F_u

抗体与抗原间亲和力用于表示抗体对抗原的识别程度, 本文使用目标函数来表示。

$$F_u = \min_{1 \leq i \neq j \leq m} \sum_{l=1}^3 W_l T_l^i \Phi_l T_l^{jT} \quad (15)$$

② 抗体与抗体间亲和力 δ_{ur}

抗体与抗体间亲和力反映了抗体之间的相似程度。本文使用 R 位连续方法计算, 即当两个抗体有超过 R 位编码相同时, 认为这两个抗体近似“相同”, 否则表示两个抗体不同。

$$\delta_{ur} = \frac{T_{ur}}{m} \quad (16)$$

其中, T_{ur} 为抗体 u 与抗体 r 中相同的编码位数, m 为抗体长度。

③ 抗体浓度 C_u

抗体浓度表示和抗体 u 相似的抗体数与抗体总数的比值, 即

$$C_u = \frac{1}{N} \sum_{r \in N} \delta_{ur} \quad (17)$$

表 1 符号及其含义
Table 1 Symbols and meanings

符号	含义	符号	含义
T_l^i	网络功能 v 的第 i 个异构体的第 l 层异构元素的特征向量	$\text{mem}(v)$	网络功能 $v \in V$ 所需的存储资源
Φ_l	第 l 层异构元素的异构矩阵	$\text{net}(v)$	网络功能 $v \in V$ 所需的网络资源
W_l	第 l 层异构元素的权重	$\text{com}(s)$	服务器 $s \in S$ 所能提供的计算资源
μ_{vs}^k	0-1 变量, 表示软件层异构元素与物理服务器之间的部署关系	$\text{mem}(s)$	服务器 $s \in S$ 所能提供的存储资源
$\text{com}(v)$	网络功能 $v \in V$ 所需的计算资源	$\text{net}(s)$	服务器 $s \in S$ 所能提供的网络资源

其中, N 为抗体总数; $\delta_{ur} = \begin{cases} 1, \delta_{ur} > Z \\ 0, \text{其他} \end{cases}$; Z 为预先设定的一个阈值。

④ 期望繁殖概率 P_u

抗体 u 的期望繁殖概率由抗体与抗原间亲和力 F_u 和抗体浓度 C_u 两部分决定, 即

$$P_u = \alpha \frac{F_u}{\sum_{u \in N} F_u} + (1 - \alpha) \frac{1/C_u}{\sum_{u \in N} 1/C_u} \quad (18)$$

其中, α 为常数。由式(18)可知, F_u 越大, 期望繁殖概率越大; C_u 越大, 期望繁殖概率越小, 既鼓励了异构度大的抗体, 又抑制了浓度高的抗体, 保证了群体的多样性。为防止目标函数最大的抗体因其浓度高而受到抑制, 导致已求得的最优解丢失, 本文采取精英保留策略, 在每次进行免疫操作之前, 将父代群中与抗原亲和度最高的和期望繁殖概率最高的若干个抗体存入记忆库。

(3) 免疫操作

① 选择

本文使用轮盘赌选择机制, 它是一种放回式随机采样方法。每个抗体进入下一代的概率就是它的期望繁殖概率, 期望繁殖概率越高, 被选进下一代的概率越大。

② 交叉

采用单点交叉法, 具体过程为: 从群体中随机选择两个抗体进行配对, 随机设置某一编码之后的位置为交叉点, 依设定的交叉概率交换两个抗体交叉点后的编码。图 6 为单点交叉示意。

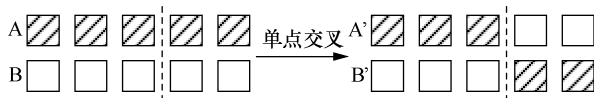


图 6 单点交叉示意

Figure 6 Single point crossover

② 变异

利用变异算子对抗体中的部分编码值进行调整, 可以从局部出发使抗体更加逼近最优解, 提高局部搜索能力, 防止“早熟”现象。本文采用常用的变异方法, 即随机选择一个抗体, 从抗体编码串中随机选择一位为变异位, 依设定的变异概率改变变异位的编码值。

具体算法步骤如算法 1 所示。

算法 1 基于免疫算法的网络功能异构部署算法

输入 物理服务器参数: 服务器数量 SN , 计算资源容量 $com(s)$, 存储资源容量 $mem(s)$, 网络资源容量 $net(s)$, 虚拟化层种类 $xkind$, 硬件层种类 $ykind$

网络功能参数: 计算资源需求 $com(v)$, 存储资源需求 $mem(v)$, 网络资源需求 $net(v)$, 软件层种类 $rkind$, 异构体数量 m 。

输出 最优部署方案 $bestdep$

1) 设置种群规模 N , 记忆库容量 $overbest$, 迭代次数 num , 交叉概率 pc , 变异概率 pm , 多样性评价参数 ps ;

2) for $i = 1 : N + overbest$ //产生初始抗体群 $chrom$, 抗体群包含父代群和记忆库两部分

3) $chrom(i,:) = randperm(SN, m)$;

4) end

5) for $j = 1 : N + overbest$ //可行性检查

6) $igj = chrom(j,:)$;

7) for $k = 1 : m$

8) 对抗体中的每一位编码进行资源容量约束检查;

9) 对不满足资源容量约束的编码进行修正;

10) 如果没有节点满足资源约束, 则部署失败

11) end

12) $chrom(j,:) = igj$;

13) end

14) for $l = 1 : num$ //迭代寻优

15) 计算 $chrom$ 中抗体的亲和度 fit ;

16) if $sum(fit) = 0$

17) 网络功能部署失败;

18) end;

19) 计算 $chrom$ 中抗体的浓度 con ; // $Z = 0.7$

20) 计算期望繁殖概率 $exc = fit / sum(fit) \cdot ps + (1 - ps) / (sum(con) \cdot con)$;

21) $[best, index] = max(fit)$; //找出最大亲和度值

22) $bestchrom = chrom(index,:)$; //找出最优抗体

23) /*形成父代群 */

24) 采用精英保留策略, 将 $chrom$ 中 fit 最高

的 3 个抗体放入父代群；

25) 将 **chrom** 中剩余抗体按照 **exc** 降序排序，取前 $N-3$ 个放入父代群；

26) 取父代群中的前 **overbest** 个抗体放入记忆库；

27) 采用轮盘赌策略对父代群进行选择操作；

28) for $ii = 1:N$

29) 对父代群中的抗体进行交叉操作；

30) end

31) for $iii = 1:N$

32) 对父代群中的抗体进行变异操作；

33) end

34) 重复步骤 5)~步骤 12)，进行可行性检查；

35) 将记忆库中的抗体加入抗体群中；

36) end

37) 更新服务器中的剩余资源；

38) **bestdep=bestchrom**； //输出最优部署方案

5 仿真结果及分析

5.1 仿真设置

为验证本文算法的有效性，在配置为 Intel Core i7-8750H CPU 2.20 GHz，8 GB 内存的计算机上，使用 Matlab 软件进行仿真。本文使用的物理网络及 SFC 请求由 GT-ITM 工具生成，为不失一般性，将计算、存储和网络资源的取值抽象为单位资源的整数倍。底层网络包含 100 个物理服务器，服务器上的计算资源容量为[50,100]的整数随机产生，存储资源容量为[100,200]的整数随机产生，网络资源为[300,500]的整数随机产生。每条 SFC 中网络功能的个数随机产生且为[3,5]的均匀分布。每一个网络功能的计算资源需求为[5,10]的整数随机产生，存储资源需求为[10,20]的整数随机产生，网络资源需求为[30,50]的整数随机产生。各层异构元素种类数设为 4。由式(2)可知，异构元素 a 和 b 之间的异构度等于非共模漏洞数量与 a 和 b 所含漏洞数量之和的比值，考虑到 Beta 分布可以通过选择合理参数适应多种分布形状，且容易获得计算结果，是表示概率的概率分布的最佳方式，本文假设异构元素之间非共模漏洞出现概率服从参数为(8,6)的 Beta 分布^[15]，图 7 为其概率分布图，可见异构元素的异

构度值主要分布在(0.2,0.9)，由此可以得到软件层、虚拟化层和硬件层的异构矩阵 Φ_1 、 Φ_2 和 Φ_3 。

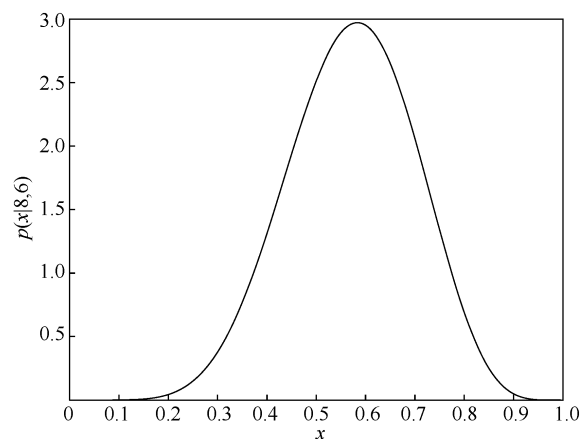


图 7 参数为(8,6)的 Beta 分布概率密度

Figure 7 Probability density plot of Beta distribution with parameters (8,6)

$$\Phi_1 = \begin{bmatrix} 0 & 0.6292 & 0.5905 & 0.5473 \\ 0.6292 & 0 & 0.8534 & 0.7707 \\ 0.5905 & 0.8534 & 0 & 0.4386 \\ 0.5473 & 0.7707 & 0.4386 & 0 \end{bmatrix}$$

$$\Phi_2 = \begin{bmatrix} 0 & 0.4728 & 0.6988 & 0.5785 \\ 0.4728 & 0 & 0.8546 & 0.4778 \\ 0.6988 & 0.8546 & 0 & 0.2399 \\ 0.5785 & 0.4778 & 0.2399 & 0 \end{bmatrix}$$

$$\Phi_3 = \begin{bmatrix} 0 & 0.3835 & 0.9345 & 0.7214 \\ 0.3835 & 0 & 0.1799 & 0.8860 \\ 0.9345 & 0.1799 & 0 & 0.3653 \\ 0.7214 & 0.8860 & 0.3653 & 0 \end{bmatrix}$$

由熵值法计算得到各层异构元素的权重为 $W_1 = 0.2880$ ， $W_2 = 0.3257$ ， $W_3 = 0.3863$ 。各层异构元素的种类为 1 到 4 的随机整数。

5.2 仿真结果分析

本文首先分析了交叉概率 P_c 、变异概率 P_m 以及种群规模 N 对算法性能的影响。图 8 为异构体数量 $m=5$ ，记忆库容量 **overbest**=10，迭代次数为 200，并且实验 200 次取平均值得到的不同控制参数下的免疫算法收敛曲线图。由图 7 可知，在 N 不变的情况下，增大 P_c 和 P_m 的取值，可以改善算法的收敛性。这是由于 P_c 和 P_m 控制着交叉和变异操作的频度，较大的交叉概率可使各代充分交叉，扩大群体分布，提高算法的全局搜索

能力；较大的变异概率可以从局部角度出发使抗体更加逼近最优解，提高算法的局部搜索能力，防止出现早熟现象。其中，当 P_c 和 P_m 从 0.1 增加至 0.5 时，改善效果较为明显；当 P_c 和 P_m 从 0.5 增加至 0.9 时改善效果相对较小。同样，群体规模 N 的大小直接影响算法的收敛性和计算效率，增大群体规模 N 可以提高算法的收敛速度，但与此同时会造成计算速度降低。

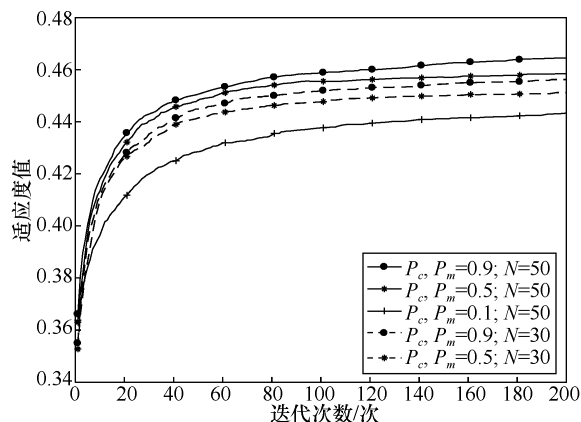


图 8 控制参数对免疫算法收敛性影响

Figure 8 The influence of control parameters on the convergence of immune algorithm

图 9 为免疫算法与遗传算法(GA)性能对比，同样异构体数量 $m=5$ ，迭代次数为 200，并且实验 200 次取平均值。从图中可以看出，免疫算法性能显著优于遗传算法。其主要原因为，遗传算法将个体适应度作为选择父代个体的唯一标准，忽视了待求解问题自身的一些特征信息或知识；免疫算法利用抗体亲和度对个体进行选择，其中包括抗体-抗原之间的亲和度和抗体-抗体之间的亲和度，在鼓励异构度大的抗体的同时抑制了浓度高的抗体，从而确保了群体的多样性。另外，观察图 9 可以发现 P_c 和 P_m 取 0.7 时遗传算法的性能优于 P_c 和 P_m 取 0.9 时，其主要原因为当 P_c 和 P_m 过大时，虽然能够产生较多的个体，增加种群的多样性，但群体中优良抗体遭到破坏的可能性增大，使搜索走向随机化。而免疫算法未发生该情况的原因在于本文采取了精英保留策略，在每次进行免疫操作之前，先将父代群中与抗原亲和度最高的和期望繁殖概率最高的若干个个体存入记忆库，防止优良抗体的丢失。

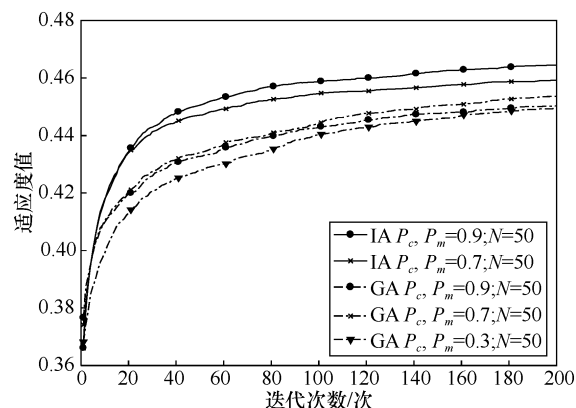


图 9 免疫算法与遗传算法性能对比

Figure 9 Performance comparison between immune algorithm and genetic algorithm

图 10 是一个箱型图，对比了异构体数量 $m=5$ 时本文提出的 minimax 算法（将异构体之间最小异构度值作为优化目标）、mean 算法（将各异构体之间异构度的均值作为优化目标）和随机部署的 random 算法的异构度离散分布情况。由图可知，minimax 算法的异构度分布相对集中，大部分在 0.5~0.6，由于该算法将最小异构度值作为优化目标，相当于对异构度取值的下限进行了约束，所以最小值为 0.473 3。mean 算法将异构度的均值作为优化目标，所以与 minimax 算法和 random 算法相比，异构度的均值是最大的，但是异构度分布相对分散，最大值高达 0.885 1，而最小值只有 0.170 0。攻击者可能会利用其中异构度较小的异构体进行协同攻击。random 算法是随机部署算法，未考虑异构度，所以异构度值的整体分布处于相对较低的水平，分布比较分散，实验效果相对较差。

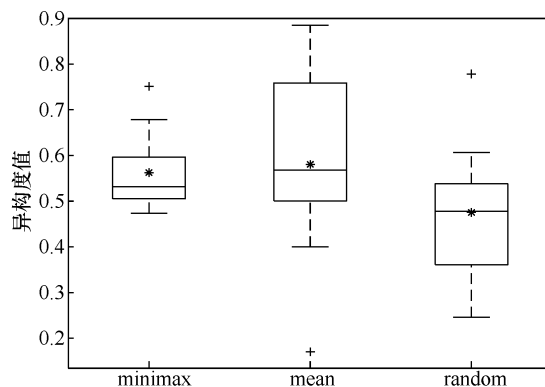


图 10 minimax、mean 和 random 异构度离散分布对比

Figure 10 Comparison of the heterogeneity discrete distribution of minimax, mean and random

图 11 为不同异构体数量下基于本文方法构建的异构体池的异构性变化情况。本文采用文献[23]中的异构性定义对网络功能异构体池的异构性进行描述。如式(19)所示, 异构性 H 由复杂性函数 Γ 和差异性函数 A 两部分构成。

$$H = \Gamma \cdot A \quad (19)$$

复杂性函数 Γ 对异构体池的多样性进行了评估, 其具体的计算公式如下。

$$\Gamma = -\sum_{i=1}^{\varepsilon} p_i \ln p_i \quad (20)$$

其中, ε 为异构体的种类数, p_i 表示第 i 种异构体在异构体池中所占的比例, $\sum_{i=1}^{\varepsilon} p_i = 1$ 。由式(20)

可知, 当组成异构体池的异构体只有一种时, Γ 为最小值 0; 当组成异构体池的异构体不同时, Γ 达到最大值 $\ln \varepsilon$ 。

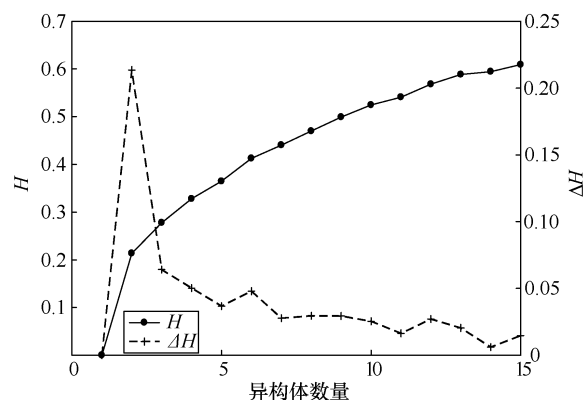


图 11 异构体池异构性与异构体数量的关系

Figure 11 The relationship between the heterogeneity of the heterogeneous entities pool and the number

异构体池的差异性量化采用 rao 二次熵^[24]的方法, 具体的计算公式如下。

$$A = \sum_{i=1}^{\varepsilon} \sum_{j=1}^{\varepsilon} Q_{ij}^2 p_i p_j \quad (21)$$

其中, Q_{ij} 为异构体 i 和异构体 j 之间的异构度, 具体取值见式(4)。由式(21)可知, 异构体之间的异构度越大, 异构体池的差异性 A 越大。

观察图 11 可以发现, 异构体池的异构性 H 与执行体数量呈正相关, 增加异构体数量可以提高 H 的取值, 增加系统多样性。但是随着异构体数量的增加, 异构体池的异构性增益 ΔH 呈下降趋

势, 与此同时部署开销越来越大, 说明增加异构体数量带来的收益越来越小。因此, 需要结合具体情况选择合适的异构体数量, 实现安全与开销的平衡。

6 结束语

网络功能虚拟化技术实现了网络功能的软硬件解耦, 云化环境使传统的物理安全边界消失, 带来许多新的安全问题。本文利用拟态防御思想提出一种网络功能虚拟化的拟态防御架构, 并针对其中异构体池的构建问题, 结合网络功能虚拟化的网络特性提出了基于免疫算法的网络功能异构冗余部署方法。结合熵值法对异构体之间的异构度进行量化评估, 以异构度的最小值为优化目标, 将网络功能异构冗余部署问题建模为极大极小值问题, 利用免疫算法对优化模型进行求解。仿真结果表明, 本文提出的方法可以在较短时间内求解出异构度较大的部署方案, 有效保证异构体之间的多样性, 增加攻击者的攻击难度, 提升系统的安全性。存在的不足是: 该方法是一种冗余部署方法, 会造成一定的部署开销增加。在接下来的工作中, 将针对部署开销与系统安全性之间的平衡问题展开进一步研究, 希望在保证系统安全性的同时尽量减少相应的开销。

参考文献:

- [1] 刘宇涛, 陈海波. 虚拟化安全: 机遇、挑战与未来[J]. 网络与信息安全学报, 2016, 2(10): 17-28.
LIU Y T, CHEN H B. Virtualization security: the good, the bad and the ugly[J]. Chinese Journal of Network and Information Security, 2016, 2(10): 17-28.
- [2] LAL S, TALEB T, DUTTA A. NFV: security threats and best practices[J]. IEEE Communications Magazine, 2017, 55(8): 211-217.
- [3] 高丽华, 董飞, 任新宇, 等. NFV 网络的安全威胁及防护方案探讨[C]// 5G 网络创新研讨会(2018)论文集. 2018.
GAO L H, DONG F, REN X Y, et al. Discussion on NFV network security threats and protection schemes[C]//Proceedings of 5G Network Innovation Seminar (2018). 2018
- [4] 吴宏建, 张如意, 芦玥. NFV 安全风险分析及应对建议[J]. 信息技术与政策, 2018, 293(11): 75-79.
WU H J, ZHANG R Y, LU Y. Analysis of NFV risks and protection concerns[J]. Information and Communication Technology and Policy, 2018, 293(11): 75-79.
- [5] 潘启润, 黄开枝, 游伟. 基于隔离等级的网络切片部署方法[J]. 网络与信息安全学报, 2020, 6(2): 96-105.
PAN Q R, HUANG K Z, YOU W. Network slicing deployment

- method based on isolation level[J]. Chinese Journal of Network and Information Security, 2020, 6(2): 96-105.
- [6] CHRISTOPHER D C, RALPH E W. Firewall system for protecting network elements connected to a public network[S]. US 2000.
- [7] LI G X, GAO W M. Research on network security system based on intrusion detection[C]//IEEE International Conference on E-business & E-government. 2010: 2096-2100.
- [8] 郭江兴. 网络空间拟态防御研究[J]. 信息安全学报, 2016, 1(4): 1-10.
- WU J X. Research on cyber mimic defense[J]. Journal of Cyber Security, 2016, 1(4): 1-10.
- [9] 张晨, 黄韬, 张娇, 等. 网络编排技术进展研究[J]. 信息通信技术, 2016(1): 68-76.
- ZHANG C, HUANG T, ZHANG J, et al. A technology research of network orchestration[J]. Information and Communications Technologies, 2016(1): 68-76.
- [10] ETSI. Network function virtualization: architectural framework[EB].
- [11] 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.
- MA H L, YI P, JIANG Y M, et al. Dynamic heterogeneous redundancy based router architecture with mimic defenses[J]. Journal of Cyber Security, 2017, 2(1): 29-42.
- [12] 全青, 张铮, 张为华, 等. 拟态防御 Web 服务器设计与实现[J]. 软件学报, 2017, 28(4): 883-897.
- TONG Q, ZHANG Z, ZHANG W H, et al. Design and implementation of mimic defense web server[J]. Journal of Software, 2017, 28(4): 883-897.
- [13] 王祺鹏, 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 P, HU H C, CHENG G Z. A DNS architecture based on-mimic security defense[J]. Acta Electronica Sinica, 2017, 45(11): 2705-2714.
- [14] 魏帅, 于洪, 顾泽宇, 等. 面向工控领域的拟态安全处理机架[J]. 信息安全学报, 2017, 2(1): 54-73.
- WEI S, YU H, GU Z Y, et al. Mimic security processor architecture oriented to the industrial control field[J]. Journal of Cyber Security, 2017, 2(1): 54-73.
- [15] 刘勤让, 林森杰, 顾泽宇. 面向拟态安全防御的异构功能等价调度算法[J]. 通信学报, 2018, 37(7): 192-202.
- LIU Q R, LIN S J, GU Z Y. Heterogeneous redundancies scheduling algorithm for mimic security defense[J]. Journal on Communications, 2018, 37(7): 192-202.
- [16] 李亮. 评价中权系数理论与方法比较[D]. 上海: 上海交通大学, 2009.
- LI L. Weights theory and weighting methods comparison in the evaluation[D]. Shanghai: Shanghai Jiao Tong University, 2009.
- [17] 公茂果, 焦李成, 杨咚咚, 等. 进化多目标优化算法研究[J]. 软件学报, 2009, 20(2): 271-289.
- GONG M G, JIAO L C, YANG D D, et al. Research on evolutionary multi-objective optimization algorithms[J]. Journal of Software, 2009, 20(2): 271-289.
- [18] DEB K. Multi-objective optimization[M]//Search Methodologies. Brilm: Springer, 2014: 403-449.
- [19] 金中. 最优化算法中极小极大问题的教学思考[J]. 创新教育研究, 2019, 7(5): 628-634.
- JIN Z. Teaching thoughts on minimax problems in optimization algorithms[J]. Innovation Education Research, 2019, 7(5): 628-634.
- [20] KHEBBACHE S, HADJI M, ZEGHLACHE D. Scalable and cost-efficient algorithms for VNF chaining and placement problem[C]//IEEE Conference Innovations in Cloud. 2017: 92-99.
- [21] 赵雨. 一类求解无约束极大极小问题的新算法[D]. 西安: 西安科技大学, 2011.
- ZHAO Y. An efficient method for solving nonlinear unconstrained min-max problem[D]. Xi'an: Xi'an University of Science and Technology, 2011.
- [22] 焦李成, 杜海峰. 人工免疫系统进展与展望[J]. 电子学报, 2003(10): 1540-1548.
- JIAO L C, DU H F. Development and prospect of the artificial immune system[J]. Acta Electronica Sinica, 2003(10): 1540-1548.
- [23] TWU P, MOSTOFI Y, EGERSTEDT M. A measure of heterogeneity in multi-agent systems[C]//IEEE 2014 American Control Conference. 2014: 3972-3977.
- [24] RAOC R. Diversity and dissimilarity coefficients: a unified approach[J]. theoretical population biology, 1982, 21(1): 30-43.

[作者简介]



张青青 (1995-), 女, 河北张家口人, 信息工程大学硕士生, 主要研究方向为网络功能虚拟化和网络安全。



汤红波 (1968-), 男, 湖北孝感人, 信息工程大学教授、博士生导师, 主要研究方向为移动通信网络和新型网络体系结构。



游伟 (1984-), 男, 江西丰城人, 信息工程大学助理研究员, 主要研究方向为新一代移动通信系统、移动通信网安全。



李英乐 (1985-), 男, 河北衡水人, 信息工程大学副研究员, 主要研究方向为移动通信网安全。