一种侧信道风险感知的虚拟节点迁移方法

黄开枝 潘启润* 袁 泉 游 伟

(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：为防御网络切片(NS)中的侧信道攻击(SCA)，现有的基于动态迁移的防御方法存在不同虚拟节点共享物理资源的条件过于松弛的问题。该文提出一种侧信道风险感知的虚拟节点迁移方法。根据侧信道攻击的实施特点，结合熵值法对虚拟节点的侧信道风险进行评估，并将服务器上偏离平均风险程度大的虚拟节点进行迁移；采用马尔科夫决策过程描述网络切片虚拟节点的迁移问题，并使用Sarsa学习算法求解出最终的迁移结果。仿真结果表明，该方法将恶意网络切片实例与其他网络切片实例隔离开，达到防御侧信道攻击的目的。

关键词：网络切片；安全隔离；侧信道攻击；马尔可夫决策过程；Sarsa学习算法

中图分类号：TP393 **文献标识码**：A **文章编号**：1009-5896(2019)09-2164-08

DOI: 10.11999/JEIT180905

A Virtual Node Migration Method for Sensing Side-channel Risk

HUANG Kaizhi PAN Qirun YUAN Quan YOU Wei

(*National Digital Switching System Engineering and Technological R&D Center*, *Zhengzhou* 450002, *China*)

Abstract: In order to defend against Side-Channel Attacks (SCA) in Network Slicing (NS), the existing defense methods based on dynamic migration have the problem that the conditions for sharing of physical resources between different virtual nodes are not strict enough, a virtual node migration method is proposed for sensing side-channel risk. According to the characteristics of SCA, the entropy method is used to evaluate the side-channel risks and migrate the virtual node from a server with large deviation from average risk. The Markov decision process is used to describe the migration of virtual nodes for network slicing, and the Sarsa learning algorithm is used to solve the optimal migration scheme. The simulation results show that this method can separates malicious network slice instances from other target network slice instances to achieve the purpose of defense side channel attacks.

Key words: Network Slicing (NS); Security isolation; Side-Channel Attacks (SCA); Markov decision process; Sarsa algorithm

1 引言

多样化的业务场景对5G网络提出了多样化的性能要求和功能要求。5G核心网应具备向业务场景适配的能力，针对5G业务场景提供恰到好处的网络控制功能和性能保证，实现按需组网的目标[1]。网络切片(Network Slicing, NS)是部署在同一物理网络上的多个相互隔离的虚拟网络，每个虚拟网络为不同的用户提供端到端的定制化服务。网络切片实例(Network Slice Instance, NSI)是一个实例化的5G核心网，在网络切片内，网络运营商可以进一步对虚拟资源进行灵活分割，按需创建子网络。

不同网络运营商的网络功能会部署在相同物理服务器中，共享物理资源(如CPU, Cache等)，这种部署方式大大提高了资源利用率，但是也给用户的隐私安全带来严重隐患。其中，侧信道攻击(Side-Channel Attacks, SCA)[2]便是利用多个NSI共享底层物理资源的特点来实施的攻击。通过分析目标NSI部署策略中存在的漏洞，攻击者可以确定目标虚拟机(Virtual Machine, VM)的位置，从而将恶意VM部署在相同的服务器上[3]。之后可以利用Prime+Probe, Flush+Reload等方法故意引起目标NSI的特定行为，同时分析共享物理资源的使用情况，探测隐私信息。文献[4]指出即使攻击者和受害者未处于同一个物理内核中，攻击者也能在3级缓

收稿日期：2018-09-20；改回日期：2019-02-26；网络出版：2019-03-11
*通信作者： 潘启润 panqirun03@163.com
基金项目：国家重点研发计划网络空间安全专项(2016YFB0801605)，国家自然科学基金创新群体项目(61521003)

Foundation Items: The National Key R & D Program Cyberspace Security Special (2016YFB0801605), The National Natural Science Foundation Innovative Groups Project of China (61521003)

存上利用SCA推测出受害者完整的密钥信息。文献[5]基于CPU负载建立侧信道，因不同VM的VCPU循环占用物理CPU，通过测量目标VM执行程序占用物理CPU的时间，便可利用特定方法且以较高成功率推断出受害者的隐私数据，且不易被检测到。

已有许多文献针对SCA问题进行研究，大致有以下3个方面：第1种是侧信道检测技术，文献[6]使用马尔可夫和贝叶斯模型来分析从Hypervisor层捕获到的信息流，完成侧信道的检测；文献[7]针对云环境中的SCA，基于事件关联机制提出了一种可以从行为角度精确定位和分析恶意侧信道的算法。该类方法将重点放在了侧信道的检测上，并没有给出具体的应对方法。第2种方法致力于通过基于体系结构、监控器或模糊时间等方法来减轻SCA带来的威胁。文献[8]设计了一种新型的Cache结构，通过动态重定位与Cache标识位来提高Cache的性能与安全；文献[9]为防御云环境中的SCA设计了一种保护系统。这类方法虽然可以防御一定类型的SCA，但并不具有通用性，且防御代价很大，增大了系统的能耗和开销。第3种方法借助VM部署和迁移的手段来防御SCA。文献[10]通过分析在应对SCA中3种常用的VM分配策略的安全性，提出了一种优先选择服务器的VM分配策略；文献[11]在迁移VM时，受害者会向攻击者实时“汇报”迁移状态，为迷惑攻击者该信息可真可假，从而提升了自身安全；文献[12]提出了一种基于安全等级的VM动态迁移方法；文献[13]将访问控制策略引入到VM的部署和迁移中，将用户之间的关系分为联盟关系和冲突关系，根据这两种关系进行VM的部署和迁移。这类方法的应用范围很广，而且虚拟化技术的灵活性使其更容易实现，但是以上方法在迁移时并未对节点的风险程度进行估计，而且在选择服务器时并未考虑与之同驻VM的SCA风险程度，导致不同NSI的VM共享资源条件不够严格。

本文针对第3种方法存在的问题，提出一种侧信道风险感知的虚拟节点迁移方法。通过分析SCA的特点，抽象出评估指标[14](操作VM的频率、Cache失效次数及VM内存利用率)并使用熵值法计算多个指标的权重系数，求出虚拟节点(即VM)的SCA的风险属性值；为严格控制不同NSI的VM共享物理资源的条件，计算服务器上承载的虚拟节点的SCA风险差异值，将偏离平均水平较大的虚拟节点进行迁移；为了提高迁移方法在随机和时变的环境中优化目标的能力，本文将虚拟节点的动态迁移问题建模成马尔可夫决策过程，并利用增强学习中的Sarsa算法求出迁移结果。仿真结果表明，本文提出的方法不但对网络服务性能产生较小影响，而且可以防御SCA，实现了恶意NSI与普通NSI的隔离。

2　问题分析

2.1　网络模型和侧信道攻击

为给用户提供端到端的定制化服务，网络运营商会针对不同的服务请求，组合不同虚拟网络功能，构建不同的NSI，多个虚拟网络功能部署在相同VM上。图1是NSI的宏观示意图，图2是部署图。网络切片管理和编排器根据服务请求信息和物理资源的状态信息，依次将多个网络切片实例G_V^k映射到物理网络G_S上。在该架构下，NSI间理论上可以利用虚拟化技术实现逻辑隔离。但诸多研究表明[2−5]，SCA能打破NSI间的逻辑隔离，窃取其他NSI的隐私信息。如图2，攻击者NSI_2为了窃取受害者NSI_1的隐私信息，首先会针对NSI_1 VM所属区域和类型，启动相同区域和类型的VM实例；然后进行VM同驻检测，查看是否和目标VM同驻成功；最后在同驻成功的服务器E上进行攻击。

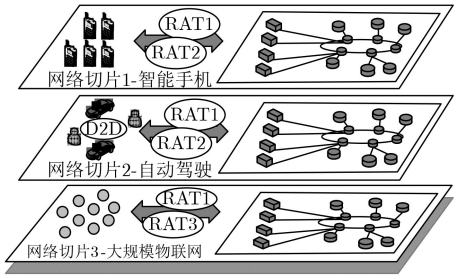

图 1 NSI宏观示意图

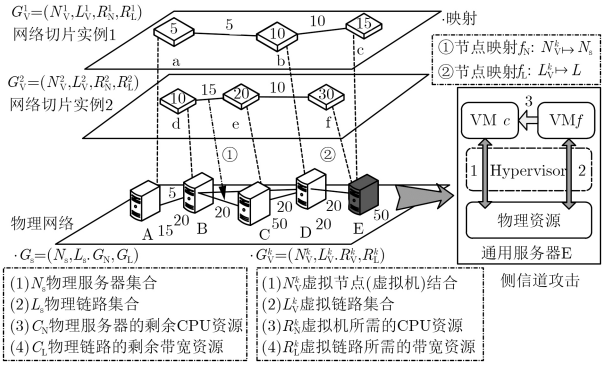

图 2 NSI部署图及侧信道攻击

2.2　问题描述

因VM c已成功和VM f 在服务器E中同驻，参照表1给出的安全影响参数[12]，可知VM c被VM f 窃取成功的信息量与二者共存的时间及VM c信息泄露的速率是相关的，故为了防御SCA，保证VM c

表 1 影响安全参数

符号	含义
ω	隐私信息泄露速率
θ	共存时间
$\mu_{\min}$	信息被成功窃取的最小信息量

安全，需满足$\omega\theta \le \mu_{\min}$。文献[12]设计的VM迁移方法就是从VM的共存时间θ着手，目的是使恶意VM在共存时间内无法获取到目标VM完整有效的隐私信息，但是该防御方法单从VM共存时间考虑，并未对SCA风险程度进行估量，故可能带来不必要的迁移。文献[15]对所有服务器进行了分组，在部署VM时依次进行组选择和组内服务器选择，在组选择和组内服务器选择时分别进行安全优化和资源优化，该方法认为部署在同组服务器上的所有VM之间发生攻击的可能很小，且并未研究如何对服务器分组，但是组划分策略直接关系防御SCA的能力。文献[16]针对VM何时迁移的问题，基于博弈论思想设计了一个迁移模型，但是它仅关注了“何时迁”，未考虑“迁到哪”，这可能会将VM迁移到SCA攻击风险很高的服务器上。

网络切片是一个开放的生态环境，不同厂商间的安全需求具有差异性，现有方法在迁移VM时并未估计其风险程度，而且在选择服务器时并未考虑与之同驻VM的风险程度，导致不同NSI的VM共享资源条件过于松弛。为此，本文提出一种侧信道风险感知的虚拟节点迁移方法，该方法从SCA攻击原理出发，根据不同VM的SCA风险值的差异程度确定出待迁移的VM，在选择服务器时，只允许风险差异值小的VM共享物理资源，并引入增强学习的思想，求出最终的迁移结果。

3 虚拟节点迁移方法

基于风险感知的虚拟节点迁移方法分以下3步：首先对虚拟节点进行SCA风险评估；然后根据SCA风险值求出服务器的风险差异值，得到待迁移虚拟节点；最后根据Sarsa算法求出的结果迁移虚拟节点。

3.1 虚拟节点风险值的评估

为NSI_k的VM定义侧信道风险值$Z_{n_v}^k(n_s, T_n)$，选取操作VM的频率、Cache失效次数和内存利用率度量SCA的风险程度。(1)操作VM的频率：操作VM的行为包括申请和撤销VM。在VM同驻检测后，若发现未能与目标NSI的VM同驻成功，便会撤销该VM，重新申请VM，再进行同驻检测。如果在观测时间内发现NSI_k操作VM频率过于频繁，便可推测NSI_k发起SCA的可能性比较大。设NSI_k在观测时间λ内操作VM次数是u_k，则操作VM的频率为$f_k = u_k/\lambda$。(2)Cache失效次数：Cache失效是指恶意VM在较短时间内连续访问相同Cache区域，出现的未访问成功的情况。NSI_k中的VM连续访问导致服务器n_sCache失效的次数为$v_k(n_s)$，若Cache失效次数太高，便可推测NSI_k发起SCA的可能性较大。(3)VM内存利用率：恶意NSI实施SCA时，需要在VM的内存中频繁进行初始化和读写内存，故NSI_k的VM内存利用率的高低可反映攻击者是否发起SCA，设NSI_k的VM内存利用率是$\eta_k(n_v)$。

网络切片是基于虚拟化技术实现的，运行在服务器上的VM均由Hypervisor管理，Hypervisor能够监控VM资源使用情况，实时探测以上3个指标的数值。为将上述指标融合求出虚拟节点风险值，采用熵值法确定各项的权重。设观测周期为T，观测间隔为λ，在一个观测周期中可得到T/λ个观测数值。观测NSI_k第i次的第j个评价指标的数值为φ_{ij}，$j \in \{1,2,3\}$, 1, 2, 3对应操作VM的频率、Cache失效次数、VM内存利用率。步骤如下：(1)对φ_{ij}进行归一化处理得到φ_{ij}'；(2)计算在第j个评价指标下第i次观测结果的比重p_{ij}；(3)计算第j项指标的熵值e_j；(4)计算第j项指标的权重w_j；(5)计算VM的SCA风险值$Z_{n_v}^k(n_s, T_n)$。

$$\varphi'_{ij} = \frac{M_j - \varphi_{ij}}{M_j - m_j},\ M_j = \max_{1\le i\le T/t}\{\varphi_{ij}\},\ m_j = \min_{1\le i\le T/t}\{\varphi_{ij}\} \tag{1}$$

$$p_{ij} = \frac{\varphi'_{ij}}{\sum_{i=1}^{n}\varphi'_{ij}},\ i = 1,2,\cdots,n, n = T/\lambda \tag{2}$$

$$e_j = -\frac{1}{\ln n}\sum_{i=1}^{n}(p_{ij}\ln p_{ij}),\ i = 1,2,\cdots,n, n = T/\lambda \tag{3}$$

$$w_j = (1-e_j)\Big/\sum_{j=1}^{3}(1-e_j) \tag{4}$$

$$Z_{n_v}^k(n_s, T_n) = \sum_{j=1}^{3}\left\{w_j\left(\frac{1}{n}\sum_{i=1}^{n}\varphi'_{ij}\right)\right\},\ i = 1,2,\cdots,n, n = T/\lambda \tag{5}$$

3.2 迁移虚拟节点的选择

当恶意NSI在目标NSI间建立了侧信道，服务提供商便违背了保障用户服务安全的服务水平协议(Service Level Agreement, SLA)，此时便可采取动态迁移的方法来防御SCA。由SCA原理知，恶意

VM的SCA风险值明显高于普通VM的SCA风险值。为了衡量同驻VM间SCA风险值的差异程度，计算服务器的SCA风险差异值。设在不超过服务器n_s资源容量的情况下，共可承载r个VM，n_s的SCA风险差异值可用式(6)表示。根据SLA设定的风险阈值Γ，在SCA风险差异值超过Γ的服务器上，将偏离平均水平较大的VM作为被迁移VM。这样不但能有针对性地迁移VM，而且当多个VM部署在同一服务器上时，风险差异值越小，则代表该服务器上不同VM侧信道攻击的风险程度越接近，此时发生SCA的可能性越小。

$$D\left(Z\left(n_s, T_n\right)\right)=\frac{1}{r}\sum_{x=1}^{r}\left(Z_{n_v}^{x}\left(n_s, T_n\right)-\frac{1}{r}\sum_{x=1}^{r}Z_{n_v}^{x}\left(n_s, T_n\right)\right)^2 \tag{6}$$

3.3 最佳迁移结果的求解

传统的VM迁移算法并未考虑VM的迁移顺序给求解结果带来的影响，为解决该问题，采用离散时间马尔可夫决策过程建模，将迁移问题看成一个序贯优化问题，基于迁移后VM同驻的状态来选择服务器。

3.3.1 迁移模型

在随机环境中，离散时间的平稳马尔可夫决策过程(Markov Decision Process, MDP)通过找出一系列符合决策者愿望的行为，最大限度提高收益，找出最佳迁移结果。

MDP用$\{S, A, r, J\}$表示：(1)S是状态空间，表示所有待迁移的VM与所有候选服务器的可能映射结果，其基本事件如式(7)，$\boldsymbol{X}(t)\in S$是t时刻的状态空间，表示t时刻的迁移状态矩阵，其中X_{ij}是$\boldsymbol{X}(t)$的一个元素，当VM i迁移至服务器j上时，值为1，否则值为0；(2)A是行为空间，其基本事件为单个VM迁移至服务器或不迁移至服务器。在实际网络环境中，节点的到达和离开符合泊松过程，由其性质可知，网络切片管理和编排器在单位时间内只能对单个VM进行迁移；(3) $r: S\times A\rightarrow R$是收益函数，指在执行行为后给予服务器的奖励。假设$t$时刻的状态为$s_t$，此时执行的行为为$a_t$，当前时刻的收益函数综合以下情况得出：(a)如果时刻t的状态s_t满足资源约束条件式(8)，即服务器上的剩余CPU资源满足迁移VM所需的CPU资源，进入(b)，否则收益为一个足够大的负实数，即$r(t)=-K_1$；(b)如果时刻t的状态s_t，在服务器上承载了来自同一NSI的两个及两个以上的VM，将收益定义为一个足够大的负实数，即$r(t)=-K_2$，否则进入(c)；(c)在时刻t的状态s_t下，计算迁移VM给整个系统带来的收益，将服务器SCA风险差异值作为当前的收益值，如式(9)所示，其中，$D\left(Z\left(n_s, T_n\right)\right)$由式(6)得出，$H_s$是当前时刻所有待迁移VM所处的服务器集合，可以看出，SCA差异值越小，收益值越大；(4)J为平均总收益函数，$J=1/T\sum_{t=0}^{T}r(t)$。

$$\boldsymbol{X}(t)=\begin{pmatrix} X_{11} & \cdots & X_{1j} & \cdots & X_{1b} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{i1} & \cdots & X_{ij} & \cdots & X_{ib} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ X_{a1} & \cdots & X_{aj} & \cdots & X_{ab} \end{pmatrix} \tag{7}$$

$$\sum_{n_v\in N_V^k}R\left(n_v\right)\le C(n_s), \forall n_s\in N_S \tag{8}$$

$$r(t)=-\sum_{n_s\in H_s}D\left(Z\left(n_s, T_n\right)\right) \tag{9}$$

3.3.2 基于Sarsa算法的虚拟节点迁移

采用增强学习的Sarsa算法来求出最终的迁移结果。如图3所示，网络切片管理和编排器将搜集到的当前网络映射情况、待迁移VM以及全网VM的SCA风险值的信息集发送至Learning Agent模块。在t+1时刻，在网络状态s_{t+1}下执行行为a_{t+1}；根据迁移后服务器SCA风险差异值计算收益函数瞬时值，之后用式(10)更新行为值函数$Q(s_t, a_t)$，其中，α_t为学习因子，γ为折扣因子；令t=t+1，在网络状态s_t下执行行为a_t，循环直至得到最优行为值$Q^*(s_t, a_t)$。通过不断学习，得到一系列有价值的行为，找到能使J最大的迁移结果，此时满足贝尔曼最优方程式(11)，且全网中所有VM只和自身SCA风险值相差不大的VM共享物理资源。

$$Q(s_t, a_t)=Q(s_t, a_t)+\alpha_t[r(s_t, a_t)+\gamma Q(s_{t+1}, a_{t+1})-Q(s_t, a_t)] \tag{10}$$

$$Q^*(s_t, a_t)=\mathrm{E}[r(s_t, a_t)+\max_{a_{t+1}}\gamma Q^*(s_{t+1}, a_{t+1})] \tag{11}$$

图4是t时刻的网络视图，设VM1, VM2是待迁移VM, S1, S2, S3是VM1的候选服务器，S4, S5是VM2的候选服务器，d代表t时刻VM的SCA风险值，D代表t时刻服务器的SCA风险差异值。在迁移

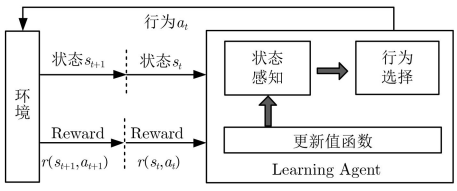

图 3 增强学习原理图

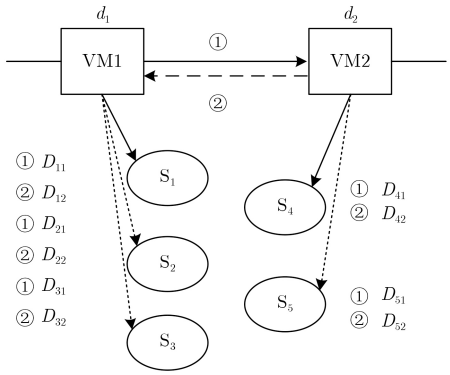

图 4 t时刻网络视图

VM时，a_t确定VM迁移顺序和迁移至的服务器。执行行为a_t后，观测下一状态s_{t+1}，然后确定t+1时刻的行为a_{t+1}，根据迁移后服务器SCA风险差异值计算收益瞬时值$r(s_t, a_t)$，然后利用式(10)更新行为值函数$Q(s_t, a_t)$，令t=t+1时刻，重复上述步骤，直至得出最佳迁移结果。

4 算法描述

算法1(见表2)是侧信道风险感知的虚拟节点迁移算法的主算法，针对已部署的NSI，首先收集操作VM的频率、Cache失效次数及VM内存利用率的数值；然后调用算法2(见表3)求出SCA风险值，并将结果返回至算法1中，接着得到待迁移VM集合，并调用算法3(见表4)求出对应的迁移位置，之后采用k短路径算法[17]映射相关链路，最后返回迁移结果。

为求出迁移结果，要不断更新Q的值，故算法的开销主要是调用Sarsa算法时产生的，在节点迁移数量较多时，路径映射算法的开销不可忽视，而其他部分的开销可忽略不计。假设有m个待迁移VM，共有n条待映射路径，v个服务器，e条物理链路。每循环1次算法，都要迁移m个VM，当行为数是r时，单次循环的复杂度是$O(mr)$，共循环了$W_{\max}$次，故复杂度是$O(mrW_{\max})$，重映射n条路径的复杂度是$O(n(e+v\lg v+k))$，故算法总复杂度为$O(mrW_{\max}+n(e+v\lg v+k))$。

5 性能评估及分析

5.1 仿真环境

在Intel(R) Pentium(R) 3.40 GHz CPU, 4 GB内存的PC机上仿真，使用GT-ITM生成物理网络和NSI请求，仿真参数如表5所示。NSI请求服从每100个时间单元到达5个的泊松分布，其生命周期服从均值为2000个时间单元的指数分布。每到达10个NSI请求，随机选择4个作为恶意切片，3个作为目标切片，3个作为普通切片。恶意VM利用同驻水印探测等技术进行目标识别，即攻击者将以较大概

表 2 算法1

输入：$G_S=(N_S, L_S, C_N, L_N)$；$G_V^k=\left(N_V^k, L_V^k, R_N^k, R_L^k\right)$；观测周期$T$；观测间隔$\lambda$；SCA风险阈值$\Gamma$

输出：最终迁移结果$M^k=\left\{M_N^k, M_L^k\right\}$

(1) for all $n_v^k \in N_V^k$ do

(2) for all $m \in [1, T/\lambda]$ do //在观测周期T内，获取T/λ个观测数值；

(3) $Y_{n_v^k \mapsto n_s}^m=\left\{f_k^m, v_k^m(n_s), \eta_k^m\left(n_v^k\right)\right\}, 1 \le m \le T/\lambda$，

$Y_{n_v^k \mapsto n_s}=\begin{pmatrix} Y_{n_v^k \mapsto n_s} \\ Y_{n_v^k \mapsto n_s}^m \end{pmatrix}$；

(4) end for

(5) $Z_{n_v^k \mapsto n_s}=\mathrm{RDV}\left(Y_{n_v^k \mapsto n_s}, T/\lambda\right)$//调用算法2(见表3)求出SCA风险值；

(6) end for

(7) 获取N_V^k所映射到的服务器集合U_S^k；

(8) for all $n_s \in U_S^k$ do

(9) 计算SCA风险值的平均值E和方差$D(Z(n_s, T_n))$；

(10) if $D(Z(n_s, T_n)) \ge \Gamma$ do

(11) 计算n_s上所有VM的风险值与均值之差，将差值最大的VM加入待迁移VM集合Ω中；

(12) end if

(13) end for

(14) $M_N^k=\mathrm{SARSA}\left(\Omega, U_S^k, G_S\right)$//调用算法3(见表4)求出VM的迁移结果；

(15) for all $n_v^k \in \Omega$ do //采用k短路径算法进行相关链路映射；

(16) for all $\chi \in \left\{n_v^k\text{所有相邻节点}\right\}$ do

(17) 获取n_v^k和χ所部署的服务器，分别为n_s^σ和n_s^χ，并求出二者之间最短路径跳数hp；

(18) 利用k短路径算法求出n_s^σ到n_s^χ且跳数为hp的路径；

(19) if $r_{\mathrm{bw}}\left(n_v^k, \chi\right) \le \min\left\{c_{\mathrm{bw}}\left(n_s^\sigma, n_s^1\right), \cdots, c_{\mathrm{bw}}\left(n_s^{\mathrm{hp}-1}, n_s^\chi\right)\right\}$ do

(20) 将$l_v\left(n_v^k, \chi\right)$映射到物理链路$l_{n_s^\sigma \to n_s \to n_s^\chi}$上，结果存入$M_L^k$；

(21) else do hp=hp+1，go to line 19；end if

(22) end for

(23) end for

(24) 返回最终迁移结果$M^k=\left\{M_N^k, M_L^k\right\}$。

表 3 基于熵值法的VM的SCA风险值求解算法(RDV)(算法2)

输入：$Y_{n_v^k \mapsto n_s}$；T/λ

输出：$Z_{n_v^k \mapsto n_s}$

(1) $\varphi_{ij}=Y_{n_v^k \mapsto n_s}$，归一化处理$\varphi$，得到$\varphi_{ij}{}'$；

(2) if $1 \le j \le 3$ do 利用式(2)计算$\varphi_{ij}{}'$的比重p_{ij}；end if

(3) if $1 \le j \le 3$ do 利用式(3)计算第j项指标的熵值e_j；end if

(4) if $1 \le j \le 3$ do 利用式(4)计算第j项指标的权重w_j；end if

(5) $Z_{n_v^k \mapsto n_s}=\sum_{j=1}^{3} w_j\left(\frac{1}{T/\lambda}\sum_{i=1}^{T/\lambda}\varphi'_{ij}\right)$，并返回该值。

表 4　基于Sarsa算法的VM迁移算法 (SARSA)(算法3)

输入：Ω；U_{S}^{k}；G_{S}

输出：M_{N}^{k}

(1) 根据Ω初始化MDP的状态空间S，初始化行为空间A，设定学习因子α_0和折扣因子γ；

(2) 令t=0，随机初始化起始状态s_0，并在空间A中随机选择行为a_0；

(3) if $1 \le m \le W_{\max}$ do //$W_{\max}$为最大循环次数

(4) 观测下一时刻状态s_{t+1}，根据行为选择策略π_Q决定时刻t+1的行为a_{t+1}；

(5) 根据式(9)计算服务器SCA风险差异值，得到收益函数瞬时值$r(s_t, a_t)$；

(6) 根据迭代式(10)更新当前行为值函数$Q(s_t, a_t)$；

(7) 令t=t+1，根据行为选择策略π_Q决定时刻t的行为a_t；

(8) if $Q^* < Q(s_t, a_t)$ do $Q^* = Q(s_t, a_t)$ end if

(9) end if

(10) 根据Q^*获得最终的迁移方案M_{N}^{k}，并返回该值。

表 5　仿真参数

参数	物理网络	网络切片实例
网络模型	Waxman	纯随机
节点连接概率p	$0.2\mathrm{e}^{-d/(0.5\times 141)}$	0.2
节点个数	100	服从$U(2, 10)$
节点和链路资源容量	服从$U(50, 100)$	服从$U(1, 20)$

率与目标VM同驻。设恶意VM掌握任意类型SCA技术，认为只要同驻成功，便可认为恶意VM具有了窃取信息的能力。学习因子设为0.2，折扣因子设为0.9，观测周期为20和30，操作VM的频率、Cache失效次数和内存利用率这3指标的数值均服从相同分布，恶意VM的值服从$U(0.6, 1)$，其他VM的值服从$U(0, 0.6)$。

5.2　结果分析

将本文方法(方法1)与文献[12]的方法(方法2)和随机迁移(方法3)[18]做对比。方法2将VM分为3个安全等级，迁移超过共存时间的VM，只允许同等级VM共享资源，且最小化迁移开销；方法3周期性地随机迁移所有VM。从SCA成功率、SCA覆盖率、请求接受率及迁移开销总和4方面对比。SCA成功率是同驻成功的VM数量比恶意NSI VM总数量；SCA覆盖率[10]指同驻成功的VM数量比目标NSI VM总数量。

如图5所示，相较于其它2种方法，方法1防御SCA的能力最强，且在观测周期较小时防御效果优于观测周期较大时的防御效果，因观测周期越小，迁移越频繁，防御SCA能力越强。方法3未评估虚拟机SCA风险，使得VM可能会迁至仍存在高SCA风险的物理节点上。方法2虽然把这些VM迁

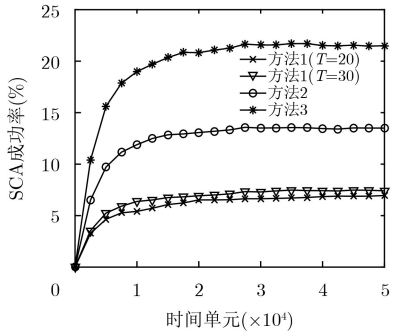

图 5 侧信道攻击成功率比较

移到同安全等级的VM上，但是它们间可能存在较大的侧信道风险差异，使得迁移后的VM仍可能与自身的SCA风险值差异大的VM共享资源，但安全等级的限制进一步缩小了发生SCA范围，使得攻击成功率不会太高。图6反映了SCA覆盖率与攻击者部署VM总数量间的关系。当攻击者部署VM总数量逐渐增多时，本文方法能使SCA覆盖率维持在一个较低水平。随着恶意VM比例的增大，本文方法基于风险值的差异程度进行迁移，可以将恶意VM集中部署在服务器上，从而将不同风险程度的VM隔离开来，降低了SCA的覆盖率，达到了防御SCA的目的。但是，在另外2种迁移策略下，随着恶意VM比例的增大，覆盖率的增长比较明显，说明其它2种方法不能达到较为理想的防御效果。采用文献[19]的方法对新到的NSI请求进行映射，然后再进行迁移。由图7可知，不同方法给网络请求

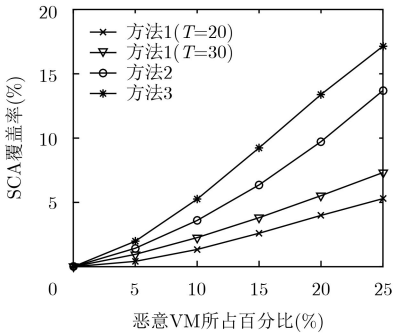

图 6 侧信道攻击覆盖率比较

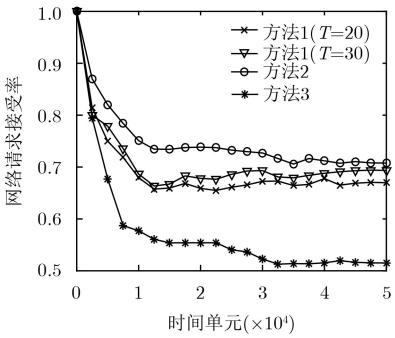

图 7 网络请求接受率比较

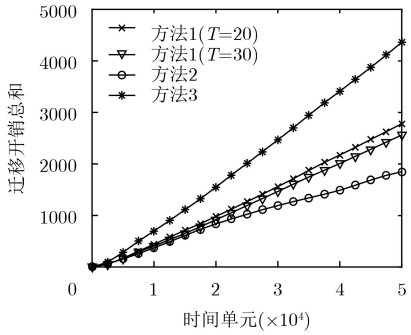

图 8 迁移开销总和比较

接受率带来了不同程度的下降。方法3迁移数量最多，对底层网络影响最大，请求接受率下降最明显。本文方法的请求接受率略低于方法2，因方法2按照安全等级进行迁移，降低了迁移节点数量，迁移的目标函数是最小化迁移开销，均衡了底层物理资源。本文迁移算法基于增强学习的思想，将VM的迁移分解成多个相互独立的VM的重映射问题，缩短了迁移算法的收敛时间，提高了物理资源在时间上的资源利用率；此外，观测周期越小，迁移越频繁，对底层物理资源的影响越大，其下降程度也越大。如图8所示，3种方法的迁移开销总和均随时间增长呈增长趋势，本文方法随时间的增长迁移开销总和并不是增长最慢的。因本文提出的方法在选择可部署的服务器时，优先考虑将待迁移的VM与SCA风险值差异差不多的VM共享物理资源，这种做法会导致求出的可部署的服务器可能与之前部署的服务器距离很远，增大了链路的迁移开销。方法2仅迁移安全等级高的VM，降低了被迁移VM数量，且迁移目标是最小化迁移开销，故迁移开销总和增长最慢。方法3被迁移的VM范围最大，资源消耗最多，迁移开销总和也增长最快。

6 结束语

针对网络切片的侧信道攻击问题，本文提出一种侧信道风险感知的虚拟节点迁移方法。从操作VM的频率、Cache失效次数及VM内存利用率3个方面，对VM的侧信道风险值进行评估，针对那些服务器上偏离平均风险程度较大的VM进行迁移，将全网风险差异总值最小作为优化目标，并使用Sarsa学习算法求出最优的迁移结果。仿真结果表明，本文方法可以防御侧信道攻击，隔离恶意网络切片实例，但是却带来了较高的迁移开销，忽略了物理资源的均衡性，在后续工作中将针对该问题进行改进。

参 考 文 献

[1] NGMN Alliance. 5G white paper[EB/OL]. https://www.ngmn.org/5g-white-paper/5g-white-paper.html, 2015.

[2] WANG Zhiming, WU Jiangxing, GUO Zehua, *et al*. Secure virtual network embedding to mitigate the risk of covert channel attacks[C]. 2016 IEEE Conference on Computer Communications Workshops, San Francisco, USA, 2016: 144–145.

[3] RISTENPART T, TROMER E, SHACHAM H, *et al*. Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds[C]. The 16th ACM Conference on Computer and Communications Security, Chicago, USA, 2009: 199–212.

[4] GULMEZOGLU B, İNCI M S, IRAZOQUI G, *et al*. Cross-VM cache attacks on AES[J]. *IEEE Transactions on Multi-Scale Computing Systems*, 2016, 2(3): 211–222. doi: 10.1109/tmscs.2016.2550438.

[5] OKAMURA K and OYAMA Y. Load-based covert channels between Xen virtual machines[C]. 2010 ACM Symposium on Applied Computing, Sierre, Switzerland, 2010: 173–180.

[6] YU Si, GUI Xiaolin, and LIN Jiancai. An approach with two-stage mode to detect cache-based side channel attacks[C]. 2013 International Conference on Information Networking, Bangkok, Thailand, 2013: 186–191.

[7] WANG Lina, LIU Weijie, KUMAR N, *et al*. A novel covert channel detection method in cloud based on XSRM and improved event association algorithm[J]. *Security and Communication Networks*, 2016, 9(16): 3543–3557. doi: 10.1002/sec.1560.

[8] WANG Zhenghong and LEE R B. A novel cache architecture with enhanced performance and security[C]. The 41st Annual IEEE/ACM International Symposium on Microarchitecture, Lake Como, Italy, 2008: 83–93.

[9] PATTUK E, KANTARCIOGLU M, LIN Zhiqiang, *et al*. Preventing cryptographic key leakage in cloud virtual machines[C]. The 23rd Usenix Conference on Security Symposium, San Diego, USA, 2014: 703–718.

[10] HAN Yi, CHAN J, ALPCAN T, *et al*. Using virtual machine allocation policies to defend against co-resident attacks in cloud computing[J]. *IEEE Transactions on Dependable and Secure Computing*, 2017, 14(1): 95–108. doi: 10.1109/tdsc.2015.2429132.

[11] ADILI M T, MOHAMMADI A, MANSHAEI M H, *et al*. A cost-effective security management for clouds: A game-theoretic deception mechanism[C]. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 2017: 98–106.

[12] 赵硕, 季新生, 毛宇星, 等. 基于安全等级的虚拟机动态迁移方法[J]. 通信学报, 2017, 38(7): 165–174. doi: 10.11959/j.issn.1000-436x.2017091.

ZHAO Shuo, JI Xinsheng, MAO Yuxing, *et al*. Research on dynamic migration of virtual machine based on security

level[J]. *Journal on Communications*, 2017, 38(7): 165–174. doi: 10.11959/j.issn.1000-436x.2017091.

[13] YU Si, GUI Xiaolin, LIN Jiancai, *et al*. A security-awareness virtual machine management scheme based on Chinese wall policy in cloud computing[J]. *The Scientific World Journal*, 2014, 2014: 805923. doi: 10.1155/2014/805923.

[14] 桂小林, 余思, 黄汝维, 等. 一种面向云计算环境侧通道攻击防御的虚拟机部署方法[P]. 中国, 102571746, 2012.
GUI Xiaolin, YU Si, HUANG Ruwei, *et al*. Virtual machine deployment method oriented to side channel attack defense of cloud computation environment[P]. CN, 102571746, 2012.

[15] LIANG Xin, GUI Xiaolin, JIAN A N, *et al*. Mitigating cloud co-resident attacks via grouping-based virtual machine placement strategy[C]. The 36th IEEE International Performance Computing and Communications Conference, San Diego, USA, 2017: 1–8.

[16] ANWAR A H, ATIA G, GUIRGUIS M. It's time to migrate! A game-theoretic framework for protecting a multi-tenant cloud against collocation attacks[C]. The 11th IEEE International Conference on Cloud Computing, San Francisco, USA, 2018: 725–731.

[17] ALJAZZAR H and LEUE S. K*: A heuristic search algorithm for finding the k shortest paths[J]. *Artificial Intelligence*, 2011, 175(18): 2129–2154. doi: 10.1016/j.artint.2011.07.003.

[18] GILLANI F, AL-SHAER E, LO S, *et al*. Agile virtualized infrastructure to proactively defend against cyber attacks[C]. 2015 IEEE Conference on Computer Communications, Hong Kong, China, 2015: 729–737.

[19] GONG Long, WEN Yonggang, ZHU Zuqing, *et al*. Toward profit-seeking virtual network embedding algorithm via global resource capacity[C]. IEEE Conference on Computer Communications, Toronto, Canada, 2014: 1–9.

黄开枝：女，1973年生，教授，博士生导师，研究方向为移动通信、无线物理层安全.

潘启润：女，1993年生，硕士生，研究方向为新一代移动通信技术、网络切片.

袁　泉：男，1991年生，博士生，研究方向为移动通信网络、网络功能虚拟化.

游　伟：男，1984年生，讲师，研究方向为密码学、5G网络安全.