doi:10.11959/j.issn.1000-436x.2016172

# CCN 中基于节点状态模型的缓存污染攻击检测算法

汤红波，郑林浩，葛国栋，袁泉
（国家数字交换系统工程技术研究中心，河南 郑州 450002）

**摘 要**：针对内容中心网络中的缓存污染攻击问题，以污染内容数量、分布状态和攻击强度 3 个参数对缓存污染攻击进行定量描述和分析，建立了攻击下的节点缓存状态模型。通过分析节点关键参数的变化，提出了基于节点状态模型的攻击检测原则，并分别以单位时间缓存替换率和请求达到率为观测参数进行算法实例化设计。仿真结果与性能分析表明，所提检测算法在应对分散式攻击与集中式攻击时，可以取得良好的检测性能。
**关键词**：内容中心网络；缓存污染攻击；缓存状态模型；攻击检测
**中图分类号**：TP393 **文献标识码**：A

# Detection algorithm for cache pollution attacks based on node state model in content centric networking

TANG Hong-bo, ZHENG Lin-hao, GE Guo-dong, YUAN Quan
(National Digital Switching System Engineering& Technological R&D Center, Zhengzhou 450002, China )

**Abstract:** Aiming at cache pollution attacks in content centric networking, the attacks were quantitatively described by three parameters, namely number of pollution contents, distribution of attack requests and attack intensity, then the cache state model of node under attack was built. Benefited from the analysis of key parameters of cache node, the attack detection principle based on node state model was put forward, correspondingly, two attack detection algorithms were instantiated with the observation parameters of cache replacement ratio and request arrival rate. The simulation results show that proposed algorithm can obtain good detection performance under each decentralized attack and centralized attack.
**Key words:** content centric networking, cache pollution attacks, cache state model, attack detection

## 1 引言

互联网自 20 世纪 90 年代以来获得了前所未有的迅猛发展，已成为数十亿用户生活中不可分割的部分。如今，互联网内容分发服务和数据流量呈现激增趋势，而网络应用的主体也逐步向内容获取和信息服务演进，用户关注的是内容信息本身以及对应的检索传输速度、服务质量和安全性，而不是从哪里获取内容[1]。

在此背景下，以信息为中心的网络结构应运而生，其中以加州大学洛杉矶分校为首开展的研究项目内容中心网络（CCN, content-centric networking）[2]最为典型。CCN 作为信息中心网络的典型代表，让内容本身成为网络通信的主体单元，采用以信息为中心的网络通信模型来支持高效的内容分发。CCN 一经提出，便得到了国内外的高度关注，被誉为最有发展潜力的结构范例，其思想也被诸多研究者广泛借鉴[3]。CCN 体系结构同 IP 网络一样都是沙漏模型，但以内容取代了 IP 地址作为其核心。通信采用“发布—请求—响应”的模式，用户只需通过名称来请求自己需要的内容（发送包含内容名称的 interest packet），当检索到所需内容后，应答数据

**收稿日期**：2015-07-21；**修回日期**：2016-06-29
基金项目：国家重点基础研究发展计划（“973”计划）基金资助项目（No.2012CB315901）；国家自然科学基金创新群体（No.61521003）；国家高技术研究发展计划（“863”计划）基金资助项目（No.2014AA01A701）
**Foundation Items:** The National Key Basic Research and Development Programs of China (973 Program) (No.2012CB315901), The Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61521003), The National High-Tech Research and Development Program of China (863 Program) (No.2014AA01A701)

（data packet）依照 interest packet 沿途留存的路径信息返回至请求者。

对于 CCN 而言，其实现高效内容分发的主要创新之处是采用广泛缓存的做法，每个 CCN 节点都带有缓存空间，用于驻留高请求频度的内容，当 data packet 沿 interest packet 的反向路径进行回传时，会按照一定策略在沿途各个节点进行缓存，当这些节点再次接收到相同内容请求时，会在缓存空间中进行检索并直接予以回传。

然而，新型网络结构的新特点往往会带来新的问题，普遍缓存的做法在提高网络效率的同时也引发了新的安全威胁，如隐私泄露、缓存污染等[4~6]，本文主要研究 CCN 中的缓存污染攻击问题，指的是攻击者通过对污染内容的请求，使其占据节点缓存空间，降低网络性能，而这些污染内容可以是正常的冷门资源，也可以由特定的恶意内容服务商提供。缓存污染攻击实际上可以视为一种 DoS 攻击[6]，但是它与传统的 DoS 攻击相比却更加灵活。首先，其具有隐蔽的特点，不需要使用洪泛的方式来攻击数据源；第二，它影响了路由节点，会使内容请求者与内容提供者双方都受到严重的影响；第三，攻击使用真实的内容占据缓存，而不是请求虚假内容，与兴趣分组洪泛攻击[6]相比更加难以检测。

## 2 问题分析

目前，提出的缓存污染攻击主要分为以下 2 类[6]：破坏内容（locality-disruption）局域分布特性和伪造内容（false-locality）局域分布特性。在 locality-disruption 攻击中，攻击者持续生成一系列 interest packet，用来请求新的污染（非流行）内容，从而破坏内容的整体分布特征，提高了污染内容的流行度，使其长期驻留于缓存之中。在 false-locality 攻击中，攻击者选择一个非流行内容的集合，周期性地对其进行请求，请求的频率应当保证不破坏当前内容的流行度分布规律，以此来伪造正常请求状态，保证攻击不被发现。

针对 locality-disruption 攻击，当前的应对手段可以分为主动检测与被动防御 2 种，前者采用各类启发式算法检测攻击，如文献[7, 8]通过检测各内容请求规律的变化来判断攻击是否发生，进而判断哪类内容更有可能是污染内容；后者则是通过设置某种规则，对某些内容不予缓存，从而降低污染内容进入缓存的概率，例如文献[9]提出了 cache shield，避免缓存流行度小于所设定阈值的内容。这些检测与防御方法在各自的场景中效果显著，但是在面对更复杂的攻击场景时却力不从心，因为一旦无法检测到内容分布特性的突变，则无法判断攻击是否发生。同样，当攻击请求较多时，cache shield 的防御效果也会下降。

针对 false-locality，文献[7,8]认为攻击者难以对正常内容的流行度进行评估，因而难以伪造，同时需要大量的攻击流量，可以通过限制突发流量来进行避免。文献[10]则认为只需要检测短时间内某端口大量突发的 interest packet，但是却忽略了存在分布式攻击的可能性。

CCN 中缓存污染攻击的本质是攻击者按照某种规律持续性地请求某些特定的污染内容，使其占据节点的缓存空间。从这一点来看，false-locality 与 locality-disruption 的分类方式仅仅是一种定性的描述，而没有定量的判断标准，若攻击者对数量可变的污染内容进行请求，这种攻击状态便可以介于所谓的 false-locality 攻击和 locality-disruption 攻击之间，比严格的 false-locality 攻击更具可行性，也比 locality-disruption 更加难以检测。对于该类攻击模式，因其动态性难以进行准确归类，而以往的工作只针对单一类型的攻击，难以应对特征不断变化的攻击行为，因此，本文抛开传统的定义方式，以定量的方式对缓存污染攻击进行描述，研究检测算法应对不同攻击时的效果，寻找应对范围更广的解决方案，主要有以下工作。

1) 对缓存污染攻击进行定量描述，并建模分析攻击下的节点缓存状态变化。

2) 提出基于节点状态模型的攻击检测原则，并给出检测算法设计示例。

3) 通过改变攻击参数设置 5 类攻击模式，据此对不同算法的检测效果做出对比分析。

## 3 缓存污染攻击建模与分析

### 3.1 攻击描述与假设

#### 3.1.1 缓存污染攻击定量描述

如问题分析所述，CCN 中缓存污染攻击的本质是攻击者按照某种规律持续性地请求某些特定的污染内容，使其长期占据节点的缓存空间[6]。由此而言，传统的 2 种攻击分类之间并没有本质的区别，都是通过类似的手段（依照某种策略请求污染内容）实现相同的目的（污染节点缓存空间）。另外，这种分类方式仅仅属于模糊性的概括，并没有给出

定量的标准，对攻击进行描述时往往不够精确。若攻击者通过调整攻击请求发送频率和所请求污染内容的数量来实现不同的攻击状态，根据传统的分类方式便难以对攻击进行准确描述，也影响了进一步的检测算法设计。

从发起方法和形式上看，缓存污染攻击的组成要素包括污染内容和攻击请求两方面。当攻击者发起攻击时，为了使污染内容能够占据节点缓存空间，其首先要请求一定数量的污染内容，其次要足够频繁地发送攻击请求分组，用以增加污染内容的驻留概率，且以不同的频率请求不同的污染内容也会达到不同的攻击效果。因此，本文以参数污染内容数量描述污染内容集合，表征缓存污染攻击的规模；分别用攻击强度和分布状态 2 个参数来刻画攻击请求发送的模式，3 个参数具体定义如下。

1) 污染内容数量 $L$：由恶意内容服务器提供或者从正常的冷门资源中选取的污染内容的总数。

2) 攻击强度 $\eta$：所有攻击请求的总到达率与正常请求到达率之比，通过比例的形式反映了攻击的强度。

3) 分布状态 $X(L)$：描述如何对这 $L$ 份污染内容进行请求，如依照 Zipf 分布[11]或者依照平均分布发送请求。

根据上述 3 个参数，缓存污染攻击可以定义为：攻击者以强度 $\eta$，依照 $X(L)$ 的分布状态，对 $L$ 份污染内容进行请求，从而使污染内容占据路由节点的缓存空间。由于每个节点的缓存空间都是有限的，一般意义上最优的攻击策略就是通过设置这些参数，使缓存空间尽可能多地被占据，同时保证攻击行为尽量隐蔽。

上述定义的描述虽然无法完全精确地刻画缓存污染攻击全部细节，但是可以清晰地描述缓存污染的主要特征，可以表征缓存污染攻击的主要行为，能够为攻击检测方法的设计提供良好的支持。它可以包括传统的 locality-disruption 或者 false-locality 攻击，当 $L$ 较大、$X(L)$ 为平均分布、$\eta$ 较小时，则属于 locality-disruption 攻击；当 $L$ 与 $\eta$ 大小适中、$X(L)$ 近似 Zipf 分布时，则属于 false-locality 攻击。

#### 3.1.2　模型假设

在定义攻击特征的 3 个参数中，攻击的分布状态可能存在的情况较为复杂，为了简化计算，从下面两方面进行分析：1) 空间分布方面，若一次攻击依照任意的分布状态，攻击者针对每类污染内容请求概率并不相同，此时对请求概率相等或相近的污染内容进行划分，可将该攻击视为多次平均分布的攻击同时叠加；2) 时间分布方面，若一次攻击中攻击请求状态随时间变化，依照同样的分析方法，可将其视为多次攻击的连续实施。因此，为了分析最一般的情况，可假定攻击依照平均分布，即针对所有污染内容的请求概率相等。对于建模对象而言，为了分析攻击的效果，则需要知道缓存节点的状态，考虑到网络边缘的接入节点直接连接网络与大量用户，受到攻击的危害最高，同时由于其与用户之间通常只有一跳，只需单节点模型便可以对其进行描述，因此将建模对象定位于网络边缘的单节点，同时这也可作为未来建立多节点模型的基础。

因此，针对网络边缘的缓存节点 $v$ 进行建模，并做出如下假设。

1) 网络中合法的提供者维护有 $N$ 份正常内容，分属于 $K$ 级不同的流行度，恶意内容源提供 $L$ 份污染内容用于污染缓存，所有内容大小相等。

2) 节点缓存空间大小为可存储 $V$ 份内容，缓存替换策略采用最近最少使用策略[12]（LRU, least recently used）。

3) 节点 $v$ 处正常请求的到达服从 Zipf 分布，对第 $k$ 级别流行度内容的请求概率为 $p(i)=\dfrac{C}{i^{a}}$，$C=(\sum\dfrac{1}{i^{\alpha}})^{-1}$，$\alpha$ =1.2[11]，其中，分别针对每级流行度内容的请求符合泊松到达并且相互独立[13]，到达率分别为 $\lambda_{i,v}$，$i=1,2,\cdots,k$，所有正常请求总到达率记为 $\lambda_{n}$。

4) 节点 $v$ 处攻击者针对所有污染内容的请求总到达率为 $\lambda_{a,v}$，分布状态为平均分布。

### 3.2　攻击下的节点缓存状态模型

使用 LRU 替换策略的缓存空间可以看作一个队列[12]：如果最近请求的一份内容在其中没有缓存，节点会将该内容的一份缓存副本保存在缓存队列的头部，同时其他所有内容依次向后移动，而最早请求的一份内容（位于缓存队列的尾部）则会移出缓存。如果最近请求的一份内容在缓存中已保留有副本，那么该内容将会重新移至队列头部，若此内容原本处于缓存队列中的第 $j$ 个位置，那么在位置 $1,\cdots,j-1$ 的所有内容依次向后转移一位。文献[13, 14]中介绍了 a-LRU 算法，对 LRU 缓存空间中的内容稳态分布进行计算，该算法可以表示为函数

$\vec{\pi}_v = contents(\vec{p}_v, |v|)$，其中，$\vec{\pi}_v = (\pi_{1,v}, \pi_{2,v}, \cdots, \pi_{N+L,v})$ 表示内容在缓存节点 $v$ 中的驻留概率，$\vec{p}_v = (p_{1,v}, p_{2,v}, \cdots, p_{N+L,v})$ 表示针对各内容请求的概率。在 a-LRU 算法的基础上对攻击下的节点缓存状态进行推导。

节点 $v$ 处请求流的到达率为正常请求与攻击请求之和，记为

$$r_v = \sum_{i=1}^{N} \lambda_{i,v} + \lambda_{a,v} \tag{1}$$

用 $\overline{\pi}_i$ 表示正常内容不在缓存中的概率，$\overline{\pi}_a$ 表示污染内容不在缓存中的概率，其值由 a-LRU 算法得出，在节点 $v$ 处，攻击下的缓存状态模型可由式(2)~式(5)进行描述。

$$m_{v,n} = \sum_{i=1}^{N} \lambda_{i,v} \overline{\pi}_i \tag{2}$$

$$P_{\mathrm{miss},n} = \frac{m_{v,n}}{\sum_{i=1}^{N} \lambda_{i,v}} \tag{3}$$

$$m_v = \sum_{i=1}^{N} \lambda_{i,v} \pi_i + \sum \lambda_a \pi_a \tag{4}$$

$$P_{\mathrm{miss}} = \frac{m_v}{\sum_{i=1}^{N} \lambda_{i,v} + \sum \lambda_a} \tag{5}$$

其中，$m_{v,n}$ 和 $m_v$ 表示正常请求的未命中流以及所有请求的未命中流，$P_{\mathrm{miss},n}$ 和 $P_{\mathrm{miss}}$ 分别表示正常请求的未命中概率以及总未命中概率。

文献[15]认为，可以通过节点的未命中概率直接反映攻击是否发生，然而从式(4)可以看出，一个缓存节点上未命中的请求流包括正常请求和攻击请求这两部分，由于攻击与正常的 interest packet 并没有区别，能观测到的未命中概率实际上是节点的总未命中概率。由于攻击请求与污染内容对节点的影响，总未命中概率并不一定与正常请求的未命中概率成正比例关系，那么如何对攻击进行检测仍是需要仔细考虑的问题，需要进一步分析攻击对节点参数的影响。

### 3.3 节点关键参数影响分析

为了分析节点遭受攻击时参数的变化，设置内容总数 $N = 10\,000$，节点缓存空间大小 $V = 500$，通过改变攻击强度 $\eta$ 以及污染内容数量 $L$ 实施不同攻击，进一步分析缓存污染攻击对节点参数的影响。

图 1 给出了缓存未命中概率随攻击强度变化的曲线，图中横坐标表示攻击强度，纵坐标表示未命中概率，各曲线从下到上分别为污染内容数量 $L$=10,50,100,150,…,1 000。图 1(a)表示总未命中概率，当污染内容数量较少时，由于污染内容以较大概率驻留于缓存中，将会导致总的未命中概率降低；当污染内容数量较多时，会产生更多的缓存替换，导致总未命中概率升高。图 1(b)为正常内容未命中概率，正常请求的未命中概率随攻击强度的增加始终递增。对比图 1(a)和 1(b)，由于攻击请求的存在，总未命中概率与正常请求未命中概率的变化趋势并不一致。

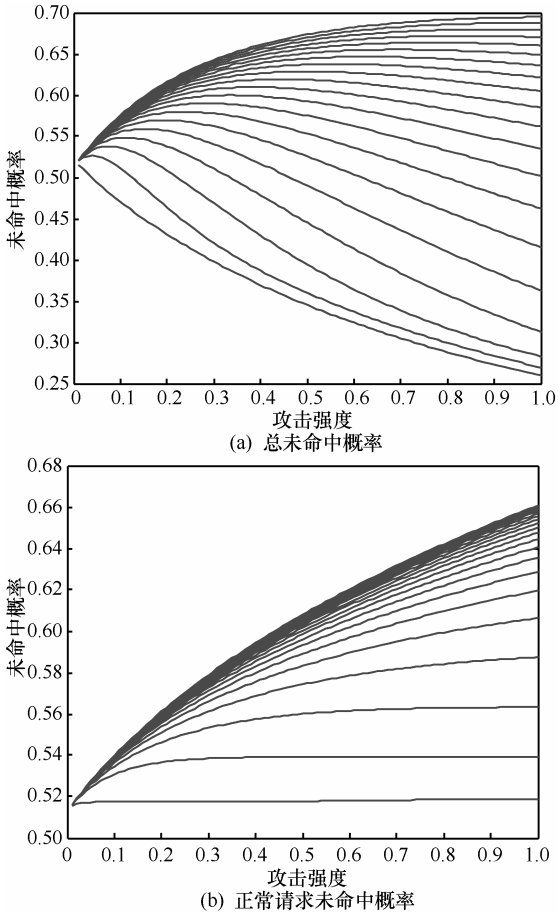



图 1 攻击对未命中概率的影响

CCN 中未命中的 interest packet 会向内容提供者处转发，当内容数据返回时缓存节点则在本地保留一份缓存副本，可以认为在稳态条件下节点的缓存空间已满，此时每有一次未命中必定会对应发生一次缓存替换。因此，缓存替换率即等价于单位时间内节点未命中的请求流。图 2 给出了节点单位时间缓存替换率随攻击强度的变化，各曲线从下到上分别为 $L$=10,50,100,150,…,1 000，通过对比图 2(a)与 2(b)可以看出，随着攻击强度的增加，正常内容替换率与总替换率主要呈递增的变化趋势，但是 2 组曲线簇的不同密集程度反映出正常内容替换率随攻击强度与污染内容数量的增加迅速上升，而总替换率变化却略为平缓。

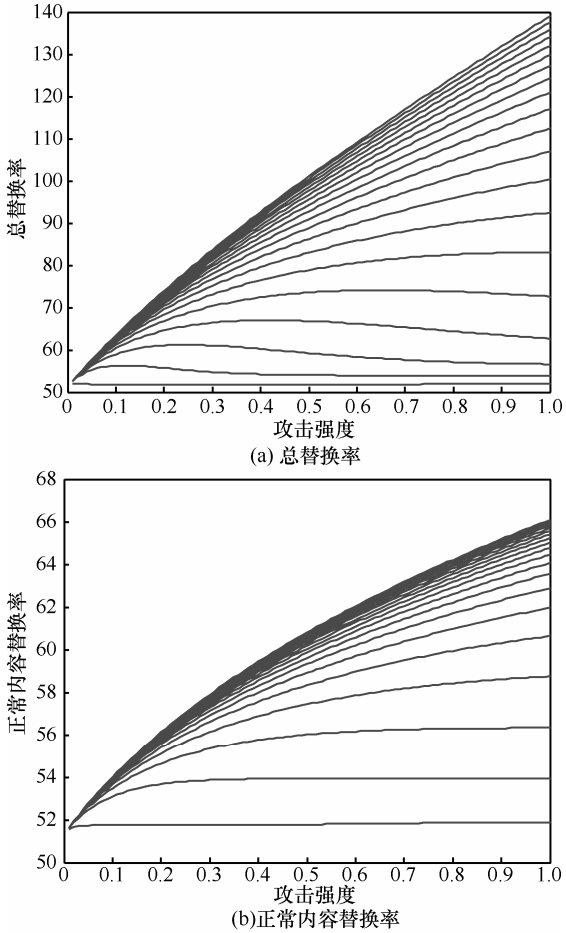



图 2　攻击对单位时间缓存替换率的影响

总而言之，当节点遭受缓存污染攻击时，不同的攻击能够对节点参数产生完全不同的影响。因此，在设计攻击检测算法时，需要重点考虑如何选择节点参数进行观测。

## 4　基于节点状态模型的攻击检测算法

通过分析节点参数变化，本节设计了基于节点状态模型的攻击检测算法，其特点在于并不严格限定具体算法内容，而是首先给出基于节点状态模型的攻击检测原则。在此原则之上，通过选择观测参数以及攻击判断标准得到实例化的算法。相比于传统检测算法的设计，这种设计方式不但可以更加清晰地体现算法设计的逻辑流程，同时能够在不影响运行的情况下随时对算法进行调整，符合未来网络灵活可变的设计需求[3]。

### 4.1　基于节点状态模型的攻击检测原则

由于受攻击者自身才能准确检测到攻击的存在[17]，从理论上来说，正常请求的未命中概率能够最准确反映攻击是否发生，但是对于缓存节点而言，其无法直接判断究竟哪一部分请求属于正常请求，因而无法直接检测正常请求流的未命中概率以及正常内容在缓存中的状态，只能根据某些间接的手段来反映是否遭到攻击。为了寻找这些“间接手段”，基于第 3 节中的模型，将节点各相关参数划分为以下 2 类。

1) 可观测参数：可以由缓存节点直接进行观察，能够直观反映节点的运行状态，例如节点上所有内容的到达率。

2) 不可观测参数：仅在模型计算中有所体现，而无法由缓存节点进行观测，例如正常请求或攻击请求的命中概率。

表 1 列举了缓存节点上的各项重要参数以及可观测性（仅限 3.2 节中模型所涉及的参数，在其他场景下仍有待补充）。

**表 1　　缓存节点 $v$ 各项参数可观测性**

| 模型参数 | 可观测性 |
|---|---|
| 总未命中概率 $P_{\text{miss}}$ | 可观测 |
| 正常请求未命中概率 $P_{\text{miss},n}$ | 不可观测 |
| 攻击请求未命中概率 $P_{\text{miss},a}$ | 不可观测 |
| 单位时间缓存替换率 $m$ | 可观测 |
| 单位时间正常内容替换率 $m_n$ | 不可观测 |
| 单位时间污染内容替换率 $m_a$ | 不可观测 |
| 内容请求的到达率 $r$ | 可观测 |
| 正常请求到达率 $\lambda_n$ | 不可观测 |
| 攻击请求到达率 $\lambda_a$ | 不可观测 |

由表 1 可以发现，不可观测参数同攻击的特征直接关联但却无法观测，而可观测参数却不能直接反映攻击是否发生。

因此，进行攻击检测所要做的就是根据缓存节点上可观测参数的变化来间接判断不可观测参数的变化。通过对前文的节点状态模型进行分析可知，这 2 类参数之间存在相互关联的关系，这便为攻击检测提供了理论依据。据此提出基于节点状态模型的攻击检测原则，主要包括以下几个方面。

1) 分析攻击的特点以及攻击可能产生的效果。

2) 选择合适的观测参数，分析其与攻击之间的关系，并据此设置攻击判断标准。

3) 若在第 2)步中选取表 1 中第 7 项 $r$ 作为检测对象，那么又可分为多种情况，因为 $r$ 可以看作一组参数 $r_i\,(i=1,2,\cdots,N+L)$，通过不同的筛选方式在 $r_i$ 中选择所需要的值便能进一步构造不同的判断方法。

4) 对观测参数的变化情况进行检测，通过与攻击判断标准的对比来判定攻击。

5) 算法实例化，根据选定的观测参数以及判断标准得到具体的算法实例。

与文献[7,8]中的设计方法进行对比，文献[7]以请求概率作为判断依据，根据请求概率的总变化量来检测攻击；文献[8]通过一系列矩阵操作在所有到达的请求中筛选出“冷门请求”，而后根据其变化来检测低强度的攻击。二者的设计方法与本节提出的设计原则有一定相通之处，这也说明了该检测原则的合理性。

### 4.2 检测算法实例化

4.1 节中列举了受攻击节点的几类可观测参数，根据模型分析，$P_{\text{miss}}$ 和 $P_{\text{miss},n}$ 之间并不成正比例关系，难以直接反映攻击。因此，分别选择单位时间缓存替换率 $m$ 或请求到达率 $r_i$ 作为观测参数，以其变化量是否超过设定的阈值为判断标准，给出基于节点状态模型的攻击检测算法实例化设计。

1) 攻击检测以一定周期进行循环，在周期 $T$ 中抽取时长 $\Delta t$，统计观测参数 $x$ 在本周期的抽样值 $x(T)$，其中，$x=m$ 或 $r_i$ $(i=1,2,\cdots,N+L)$，$m$ 和 $r_i$ 分别为单位时间缓存替换率和每类请求的到达率。

2) 由式(6)计算检测结果，即观测参数抽样值相比上周期的变化量 $\delta_x(T)$，并按照式(7)取 $\delta_r(T)$ 为所有 $\delta_{r_i}(T)$ 中最大值，用来表示请求到达率变化最大者。

$$\delta_x(T)=x(T)-x(T-1),$$
$$x=m \text{ 或 } r_i, i=1,2,\cdots,N+L \tag{6}$$
$$\delta_r(T)=\max(\delta_{r_i}(T)) \tag{7}$$

3) 根据 $\delta_x$ 在 $n$ 个周期内变化的标准差设置阈值 $\tau$，如式(8)所示，其中，$\tau_{\max}$ 表示可接受的最大阈值。若 $\delta_x$ 在某周期内的变化超过阈值 $\tau$，则判定攻击发生。

$$\tau=\min\left(\frac{\sum_{t=T-n}^{T}\delta_x(t)}{n}+\sqrt{\frac{1}{n}\sum_{t=T-n}^{T}\left(\delta_x(t)-\frac{\sum_{t=T-n}^{T}\delta_x(t)}{n}\right)^2},\tau_{\max}\right) \tag{8}$$

给出攻击检测算法如下。

**输入** *time_interval*：当前时间

$t$：周期时长

$\Delta t$：采样时长

**输出** result_of_detection：是否发生攻击

```
1)  result_of_detection=false
2)  if (time_interval < Δt )
3)      根据式(6)计算 δ_x
4)      if( δ_x < τ )
5)          In_Queue( δ_x ) //记录当前的 δ_x
6)          根据式(7)更新 δ_x
7)      else if ( δ_x > τ )
8)          result_of_detection =true
9)      end if
10) else if (time_interval > t )
11)     time_interval=0 // 周期结束，重新计时
12) end if
13) return result_of_detection
```

可以看出，该算法设计思路重点在于“选择”而不是“设计”，即在什么场景下选择什么样的检测方法更加合适。因此，实际操作时应重点分析不同算法在不同场景下的适用性，而不是讨论相同场景下同一类算法的优劣。

## 5 仿真与性能分析

实际中可能存在模式繁多的缓存污染攻击，由于篇幅的限制，很难对所有的攻击模式进行遍历。为了能够较全面地检验算法的性能，依据污染内容数量的多少和攻击强度的大小，将缓存污染攻击分为 5 类，通过对每一类设定具有代表性的参数，得到典型的检测效果，用以反映对不同特征攻击的检测性能。

由于当前 CCN 仍然处于理论研究和实验验证阶段，缺乏实际的网络运营数据。在不失一般性的前提下，这里按照独立参考模型生成请求序列。设置正常内容总数 $N=10\,000$，正常请求依照 Zipf 规律发送（$\alpha=1.2$）[11]，interest packet 发送频率为 1 000 个/秒，节点缓存空间大小 $V=500$，一个周期的时间设置为 60 s，检测算法的参数统计范围为 10 周期，对节点的攻击从第 20 周期开始，通过改变污染内容数量 $L$ 和攻击强度 $\eta$ 设定 5 类攻击模式。

1) 轻量攻击：$L$=100，$\eta=0.1$，此时污染内容数量较少，攻击强度较小，表现为以较低的频率对少数内容进行请求，虽然对网络性能影响较小，但可能属于内容的违规发布，即通过定期请求某些违规内容使其驻留于缓存中，即使服务器被屏蔽，缓存中的内容依然对用户可达。

2) 集中式攻击：$L$=100，$\eta = 0.9$，此时污染内容数量较少，攻击强度较大，表现为对少数内容进行集中请求，可保证内容以极大概率占据缓存。

3) 分散式轻量攻击：$L$=1 000，$\eta = 0.1$，此时污染内容数量较多，攻击强度较小，表现为对大量污染内容进行请求，但针对每类内容的请求到达率都极低，这种状态类似于普通的 locality-disruption 攻击，即不断请求新的非流行内容。

4) 分散式攻击：$L$=1 000，$\eta = 0.9$，此时污染内容数量较多，攻击强度较大，表现为对大量污染内容的大量请求，更类似于传统 DoS 攻击方式，无论对链路还是节点都影响较大。

5) 中量攻击：$L$=500，$\eta = 0.5$，污染内容数量与攻击强度均处于中等水平，攻击状态介于上述 4 类之间，类似于 false-locality 攻击。

文献[17]提出了一种检测机制 LWM (light weight mechanism)，其所观测的参数是所有请求概率变化量的总和，通过检测参数突变来判断攻击是否发生，检测效果与污染内容数量无关，只与攻击强度成正比。文献[7]提出 CUSUM 算法，将节点可能缓存的所有内容映射到一个矩阵中，通过异或的操作筛选出其中的非流行内容，而后通过矩阵秩（rank）的变化来判断攻击，此处使用 100×100 的矩阵记录节点可能存储的非流行内容，以该矩阵的秩作为观测参数，秩越大表示对冷门资源的请求越多，即受到攻击的可能越大。将 4.2 节中的 2 种实例化检测算法记为单位时间缓存替换率检测和请求到达率检测，连同 LWM 以及 CUSUM 共 4 种检测算法分别对上述 5 类攻击进行检测。

图 3 给出了 4 种算法针对轻量攻击的检测情况，以检测结果的归一化数值（即归一化 $\delta$ 值，定义为 $\frac{\delta}{\tau}-1$）来衡量检测的效果，其大于 0 表示可检测到攻击，值越大表示检测效果越好。由于轻量攻击特征不明显，同时对网络的影响很小，通过检测参数变化的方式难以进行识别，只有根据请求到达率检测可以实现归一化 $\delta$ 值大于 0，但其最大值仅为 0.10。总体来看检测效果均较差。

图 4 给出了 4 种算法针对集中式攻击检测结果的归一化数值，由于请求到达率检测选取的观测参数为请求率变化的最大值，对于集中式攻击十分敏感，其归一化 $\delta$ 值可以高达 8。该类攻击强度较大，LWM 也可取得较好的检测效果，归一化 $\delta$ 值为 2.05。单位时间替换率检测的归一化 $\delta$ 值为 0.15，CUSUM 则无法进行检测。前 3 种检测算法均基于历史统计判断攻击，当攻击持续时，旧的统计数据逐渐被替换，导致曲线呈下降趋势。

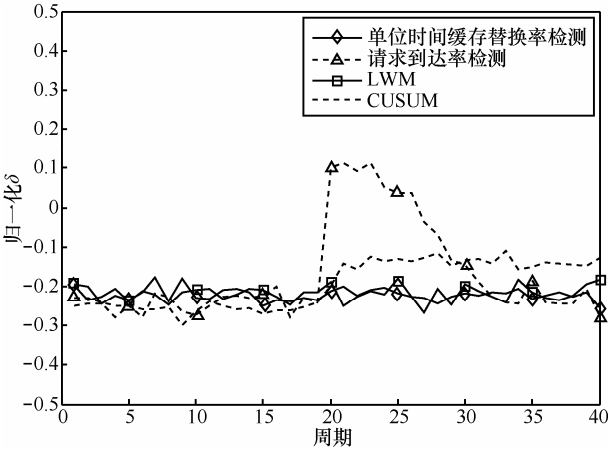



图 3　针对轻量攻击（第一类攻击）的检查结果

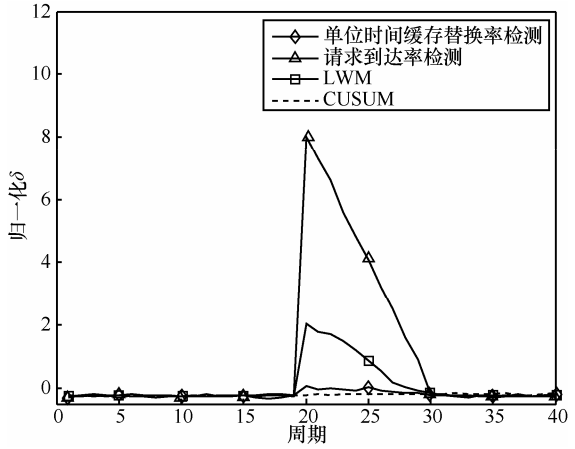



图 4　针对集中式攻击（第二类攻击）的检查结果

图 5 给出了 4 种算法针对分散式轻量攻击检测结果的归一化数值，CUMSUM 算法通过映射的方式对所有冷门请求进行统计，冷门请求越多则检测效果越好，归一化 $\delta$ 值可达 2.80，另外算法中将变化的参照量以及阈值设置为固定值，而不是基于历史统计数据得出，所以检测曲线并不会因统计数据的变化而下降。对于其他 3 种算法而言，由于该类攻击强度较低，单独针对每份污染内容的请求到达率均极低，难以实现有效检测。

图 6 给出了 4 种算法针对分散式攻击检测结果的归一化数值，该类攻击特征最为明显，导致网络状态的变化最大，因此各算法均对其有一定效果。

其中，LWM 算法效果最好，归一化$\delta$值为 2.01。单位时间缓存替换率检测效果次之，归一化$\delta$值为 1.30。另外 2 种检测算法效果较差，但也可实现归一化$\delta$值大于 0。

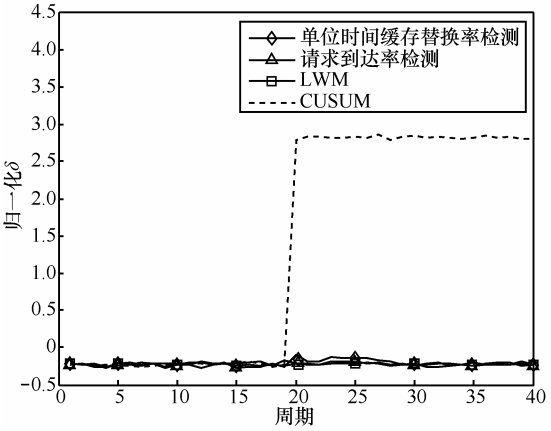



图 5 针对分散式轻量攻击（第三类攻击）的检查结果

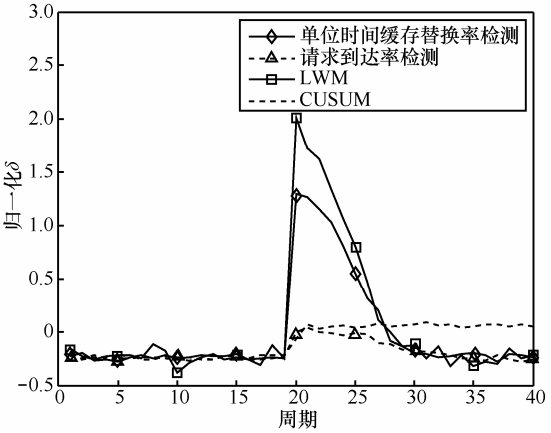



图 6 针对分散式攻击的检查结果

图 7 给出了 4 种算法针对中量攻击检测结果的归一化数值，由于中量攻击状态介于上述 4 种之间，

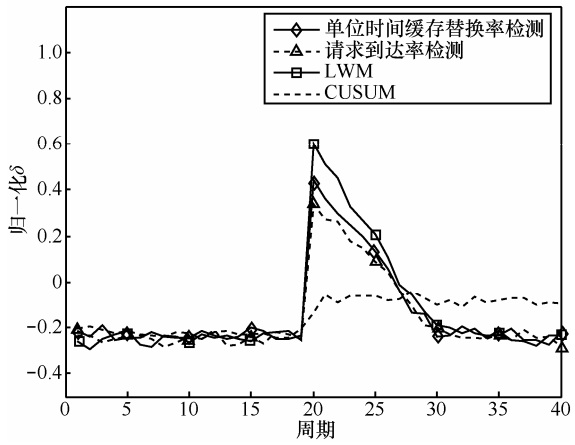



图 7 针对中量攻击的检查结果

因此各类检测算法均对其有一定效果，但总体来说检测效果并不好。其中，LWM 算法检测所得归一化$\delta$值最高，但仅为 0.6，因此并没有一种算法能够特别有效地针对此类攻击。

表 2 列出各算法检测结果的归一化数值，反映了上述 4 种检测算法适用范围各不相同。LWM 的检测范围最广，可以检测第二、四、五类攻击，但由于网络中正常请求的实时变化对其观测参数的影响较大，实际应用中误判的概率也较大；基于单位时间替换率的检测算法适用范围与 LWM 类似，虽然结果的归一化值较小，但由于其观测参数较单一，误判率相比 LWM 较小。其余 2 种应对面稍窄，请求到达率检测方法对于第二类攻击最敏感，归一化$\delta$值可高达 8.00，同时对第五类攻击也有一定效果；而 CUSUM 算法仅仅适用于检测第三类攻击。综合 4 种检测方法，可以对第二、三、四类攻击进行有效检测，而对第五类攻击的检测效果均较为一般。第一类攻击虽然难以检测，但可以被文献[9]提出的 cache shield 所拦截。总之，不同的检测算法对于不同攻击的检测效果不同，本文提出的 2 种检测算法实例能够与现有的检测算法进行互补，在实际应用中还需根据情况进行具体设计，此外由于第 5 类攻击特征不明显，导致各类算法对其检测效果较为一般，也应对其多加考虑。

**表 2 检测结果的归一化数值**

| 检测方法 | 攻击模式 | | | | |
|---|---|---|---|---|---|
| | 轻量攻击 | 集中式攻击 | 分散式轻量攻击 | 分散式攻击 | 中量攻击 |
| 单位时间替换率检测 | −0.22 | 0.15 | −0.10 | 1.30 | 0.44 |
| 请求到达率检测 | 0.10 | 8.00 | −0.21 | −0.02 | 0.34 |
| LWM | −0.17 | 2.05 | −0.22 | 2.01 | 0.60 |
| CUSUM | −0.18 | −0.20 | 2.80 | 0.10 | −0.05 |

## 6 结束语

本文对 CCN 中的缓存污染攻击进行研究，指出了传统模糊式的攻击分类方式并不能准确描述 CCN 缓存污染攻击的状态，进而导致了攻击检测算法无法实现全面的检测与防护。为此，本文对 CCN 中的缓存污染攻击重新进行定量描述，以污染内容数量、分布状态、攻击强度这 3 个参数描述攻击的特征，建立攻击下的节点缓存状态模型分析不同攻击对节点的影响。提出了基于节点状态模型的攻击检测原则，并分别以单位时间缓存替换率和请求到达率作为观测参数进行算法实例化。仿真结果表

明，本文设计的 2 种检测算法实例能够与现有的检测算法进行有效互补，在实际的应用中还需结合不同检测算法的适用范围进行权衡决策。下一步的工作包括建立多节点网络模型，并考虑攻击请求的相关性和多样的分布状态，进一步验证算法的适用性。

## 参考文献：


[1] XYLOMENOS G, VERVERIDIS C, SIRIS V, et al. A survey of information-centric networking research[J]. IEEE Communications Surveys & Tutorials, 2014, 16: 1024-1049.

[2] JACOBSON V, SMETTERS D K, THORNTON J D, et al. Networking named content[J]. Communications of the ACM, 2012, 55(1): 117-124.

[3] 兰巨龙, 程东年, 胡宇翔. 可重构信息通信基础网络体系研究[J]. 通信学报, 2014, 35(1): 128-139.
LAN J L, CHENG D N, HU Y X. Research on reconfigurable information communication based network architecture[J]. Journal on Communications, 2014, 35(1): 128-139.

[4] ACS G, CONTI M, GASTI P, et al. Cache privacy in named-data networking[C]//IEEE International Conference on Distributed Computing Systems. Philadelphia, USA, 2013: 41-51.

[5] CHAABANE A, DE CRISTOFARO E, KAAFAR M A, et al. Privacy in content-oriented networking: threats and countermeasures[J]. ACM SIGCOMM Computer Communication Review, 2013, 43(3): 25-33.

[6] LAUINGER T. Security & scalability of content-centric networking[D]. TU Darmstadt, 2010.

[7] CONTI M, GASTI P, TEOLI M. A lightweight mechanism for detection of cache pollution attacks in named data networking[J]. Computer Networks, 2013, 57(16): 3178-3191.

[8] PARK H, WIDJAJA I, LEE H. Detection of cache pollution attacks using randomness checks[C]//IEEE International Conference on Communications (ICC). Ottawa, 2012: 1096-1100.

[9] XIE M, WIDJAJA I, WANG H. Enhancing cache robustness for content-centric networking[C]// IEEE INFOCOM Annual IEEE International Conference on Computer Communications. Orlando, 2012: 2426-2434.

[10] SANDBERG A, EKLÖV D, HAGERSTEN E. Reducing cache pollution through detection and elimination of non-temporal memory accesses[C]//2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis. Washington DC: IEEE Computer Society, 2010: 1-11.

[11] KIM Y, YEOM I. Performance analysis of in-network caching for content-centric networking[J]. Computer Networks, 2013, 57(13): 2465-2482.

[12] DAN A, TOWSLEY D. An approximate analysis of the LRU and FIFO buffer replacement schemes[M]. New York, USA: ACM Publisher, 1990: 143-152.

[13] CHAI W K, HE D, PSARAS I, et al. Cache “less for more” in information-centric networks[C]//IFIP Networking. Prague, Czech, 2012: 27-40.

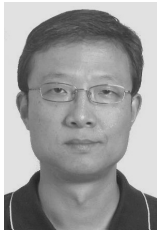
[14] ROSENSWEIG E J, KUROSE J, TOWSLEY D. Approximate models for general cache networks[C]// IEEE INFOCOM 2010. San Diego, 2010: 1-9.

[15] DENG L, GAO Y, CHEN Y, et al. Pollution attacks and defenses for Internet caching systems[J]. Computer Networks, 2008, 52(5): 935-956.

[16] MOHAISEN A, ZHANG X W, SCHUCHARD M, et al. Protecting access privacy of cached contents in information centric networks[C]// ACM SIGSAC Symposium on Information, Computer and Communications Security. Hangzhou, China, 2013: 173-178.

[17] AFANASYEV A, MAHADEVAN P, MOISEENKO I, et al. Interest flooding attack and countermeasures in named data networking[C]//IFIP Networking Conference. New York, 2013: 1-9.


## 作者简介：

**汤红波**（1968-），男，湖北孝感人，国家数字交换系统工程技术研究中心教授，主要研究方向为移动通信网络与安全、未来网络体系结构、内容中心网络。

**郑林浩**（1991-），男，河南辉县人，国家数字交换系统工程技术研究中心硕士生，主要研究方向为内容中心网络、网络安全。

**葛国栋**（1985-），男，陕西咸阳人，国家数字交换系统工程技术研究中心工程师，主要研究方向为未来网络体系结构、网络安全。

**袁泉**（1991-），男，山东青岛人，国家数字交换系统工程技术研究中心硕士生，主要研究方向为未来网络体系结构、移动通信。