软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2016,27(12):3085−3103 [doi: 10.13328/j.cnki.jos.004938]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

无线传感器网络中虫洞攻击实时被动式探测*

鲁 力[1], Muhammad Jawad HUSSAIN[1,2], 朱金奇[3]

[1](电子科技大学 计算机科学与工程学院,四川 成都 611731)

[2](电子科技大学 通信与信息工程学院,四川 成都 611731)

[3](天津师范大学 计算机与信息工程学院,天津 300387)

通讯作者: 鲁力, E-mail: luli2009@uestc.edu.cn

摘 要: 在无线传感器网络所面临的安全问题中,虫洞攻击是最严重的威胁之一.由于无线传感器节点的资源非常有限,因此,适用于有线网络上的基于密码学的安全技术不能直接移植于无线传感网络.目前已知的传感网中,虫洞攻击的探测方案在应用上存在问题,这些方案或需要精确时间同步、或额外的定位算法或硬件、或有较大的通信开销,并且,现有方案均不能检测可自适应调整攻击策略的主动虫洞敌手.结合无线传感器网络的特点,提出了基于拓扑的被动式实时虫洞攻击探测方案,称为 Pworm.通过利用虫洞攻击的主要特征——大量吸引网络流量和显著缩短平均网络路径,Pworm 不需要任何额外的硬件,只需要收集网络中部分路由信息,就能实时地探测虫洞节点,即使是主动虫洞节点,也不能通过改变自身攻击策略而躲避探测.实验结果和分析表明:该方案具有轻量级、低漏报率、高可扩展性等优点,适用于大规模无线传感网络.

关键词: 无线传感器网络;虫洞攻击;实时被动检测

中图法分类号: TP393

中文引用格式: 鲁力,Hussain MJ,朱金奇.无线传感器网络中虫洞攻击实时被动式探测.软件学报,2016,27(12):3085−3103. http://www.jos.org.cn/1000-9825/4938.htm

英文引用格式: Lu L, Hussain MJ, Zhu JQ. Real-Time and passive wormhole detection for wireless sensor networks. Ruan Jian Xue Bao/Journal of Software, 2016,27(12):3085−3103 (in Chinese). http://www.jos.org.cn/1000-9825/4938.htm

Real-Time and Passive Wormhole Detection for Wireless Sensor Networks

LU Li[1], Muhammad Jawad HUSSAIN[1,2], ZHU Jin-Qi[3]

[1](School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

[2](School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

[3](School of Computer and Information Engineering, Tianjin Normal University, Tianjin 300387, China)

Abstract: Wormhole attack is one of the severe threats to wireless sensor and ad hoc networks. Most of the existing countermeasures either demand high network overheads or require specialized hardware to capture the specific symptoms induced by the wormholes, which in result, limits their applicability. This paper exploits an inevitable symptoms of wormhole attack and proposes Pworm, a real-time and passive wormhole detection and localization scheme based on the key observation that a large amount of network traffic are attracted by the wormholes. The proposed scheme can silently observe the variations in network topology to infer the existence of wormholes. Besides the scheme solely depends on network routing information and does not demand any specialized hardware. System performance of the

* 基金项目: 国家自然科学基金(61472068, 61173171, 61103227); 中国博士后基金(2014M550466)

Foundation item: National Natural Science Foundation of China (61472068, 61173171, 61103227); Postdoctoral Science Foundation of China (2014M550466)

收稿时间: 2015-06-18; 修改时间: 2015-07-27; 采用时间: 2015-11-02; jos 在线出版时间: 2015-11-18

CNKI 网络优先出版: 2015-11-18 14:58:48, http://www.cnki.net/kcms/detail/11.2560.TP.20151118.1458.003.html

scheme is evaluated through extensive simulations of 100 to 500 nodes for various network scales and the results show that Pworm is well suited for false alarms with good scalability and low time delay.

Key words: wireless sensor networks; wormhole attack; real-time and passive detection

虫洞攻击是无线传感器网络中最具威胁的攻击之一.如图 1 所示,在虫洞攻击中,虫洞节点 n_1 和 n_2 通过高速带外信道将数据包发送到距离很远的节点.由于无线传感器网络中,传感器节点一般采用最短路径路由,因此,虫洞节点可以吸引网络中大量流量,并通过控制网络流量发动一系列攻击,如丢包、恶意篡改包的内容等.更严重的是,虫洞节点吸引大流量数据包的特点,为一系列更具威胁的攻击提供了实施基础,如中间人攻击、密钥破解、协议逆向工程等.

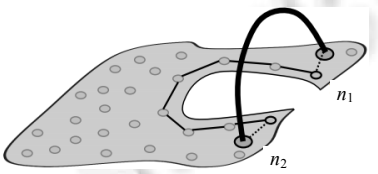


Fig.1 Wormhole attack model

图 1 虫洞攻击模型

本文主要集中分析两种类型的虫洞敌手模型:主动敌手和被动敌手.被动敌手指敌手利用高速信道持续吸引和转发流量;而主动敌手会主动感知网络状态的变化,并据此调整攻击策略.例如网络控制端,如 Sink,在开始虫洞探测之前,会在网络中广播一个控制包以收集网络拓扑.主动敌手可将该控制包作为探测的标识从而改变攻击策略,如立即停止攻击,以躲避探测.另外,这两种虫洞敌手都能捕获并完全复制合法节点.

目前,学术界已经提出了很多针对传感器网络虫洞攻击的探测方法.但是,已知的方法有的需要给节点增加额外的特殊硬件,有的则需要做很强的假设.例如,有些方法中需要知道节点的地理位置,所以提出在传感器节点上加装诸如 GPS、定向天线等特殊的装置,但这会给系统带来大量的额外硬件开销;另一类方法基于对网络的强假设.例如,需要精确的全局时钟同步、特殊的定位节点或探测事件.这些需求和假设极大地限制了这些探测方法在无线传感器网络中的应用.另外,通过临时停止攻击或中止高速连接等方式,主动虫洞敌手可有效躲避现有方法的检测.

本文提出了被动式的实时虫洞检测方案(passive wormhole detection,简称 Pworm),以发现并定位无线传感器网络中的虫洞敌手,包括主动和被动敌手.Pworm 的基本思想基于一个重要观察:网络中,普通节点总是试图选择到 Sink 节点(数据接收节点或控制节点)最短的路径.因此,虫洞链路可以利用高速信道缩短网络平均路径,并吸引网络中大量流量.另外,虫洞节点在网络所有路径中的出现比例远远高于其他普通节点.基于以上观察,通过计算网络中所有节点在路径中的出现频率和路径长度的变化,Pworm 可确定虫洞攻击的发生并定位虫洞.为了让虫洞敌手不能感知探测过程,我们提出了将网络路径(或拓扑)收集过程通过创新的安全包标记算法融入到网络正常收发数据包的运行过程中.而虫洞节点要发起攻击就必须建立虫洞连接来转发数据包,此时, Sink 节点可以通过虫洞连结导致的网络路径变化检测其存在.若虫洞节点要避开探测,就需要关闭虫洞连结,从而无法吸引网络流量发起攻击.

Pworm 由两部分组成:网络拓扑收集和虫洞攻击探测.拓扑收集阶段,我们提出了一种安全包标记算法实时获取网络拓扑,该算法可以抵抗虫洞敌手对数据包的篡改和丢弃;在虫洞探测阶段,网络中的控制端(如 Sink)分析网络拓扑中的路径变化以确定虫洞攻击是否出现,并且快速定位虫洞节点.Pworm 有 3 个主要优点.

(1) 被动式探测

被动式探测指不需要网络发起任何特殊的探测事件或行为,使得敌手可以察觉到虫洞探测正在或将要执行,也就是说在被动式探测中,PWorm 将探测过程融入网络正常收发数据包的过程中,使得敌手感知不到自己正在被探测.具体而言,在 Pworm 中,Sink 通过分析网络拓扑来探测并定位虫洞,而网络拓扑信息通过安全包标记

方法融入网络数据包的转发过程中,即,虫洞探测过程与网络正常运行无法区分,使得虫洞节点,即使能自适应调整攻击策略的主动虫洞也不能躲避探测,从而实现了被动式探测的功能.

(2) 轻量级探测

在通信方面,Pworm 采用的安全包标记算法仅需在数据包中增添少量字节,并且无需对每个数据包进行标记就可实现拓扑收集.在计算方面,虫洞探测和定位发生于 Sink 端而非各普通节点,不会给网络中普通节点带来计算开销和增添任何额外硬件.

(3) 实时探测

由于将虫洞探测过程融入网络正常工作中,当 Sink 实时检测到网络拓扑变化时,就可以快速发现虫洞攻击并定位虫洞节点.需要承认的是:Pworm 是一个概率方法,当虫洞节点不工作或仅吸引少量流量时会产生漏报.但如果虫洞吸引的流量越少,那它对网络流量的影响就越低.

综上所述,本文主要贡献包括以下两点:(1) 在资源受限的无线传感器网络中,提出了轻量级、被动式的实时虫洞探测和定位方法;(2) 能探测主动虫洞敌手,通过利用安全包标记算法将虫洞探测过程融入网络数据包转发中,使得探测过程在网络运行阶段持续进行,让主动虫洞敌手无法感知探测事件,从而不能自适应地调整其攻击策略.

本文首先介绍相关工作,并描述系统模型(包括敌手模型)和设计目标.详细介绍方案设计后,对方案的安全性进行了分析.通过实验验证了本文方案的可行性和性能.最后总结全文.

1 相关工作

目前,对虫洞攻击的探测方法可分为以下几类.

- 基于节点位置

某些方法需要特殊的定位硬件来发现虫洞造成的距离不匹配现象.Capkun 等人[1]提出了一个名为 SECTOR 的协议,通过特殊的测距硬件精确地测量节点之间的距离.文献[2]提出了一种基于空间和时间数据包限制的方法.该方法需要在网络中的节点上添加 GPS 模块,以进行节点精确定位并探测虫洞节点.这些方法都需要在网络节点上增加额外硬件,带来较大的成本开销,不适用于已部署的传感器网络.

Ban 等人[3]提出:利用在虫洞连接两侧虫洞节点会将网络节点分割为两个子集,每个子集中的节点到另一子集节点的距离在没有虫洞连接时大于虫洞连接出现时的距离,可以据此来探测虫洞节点.但该方法只适用于被动虫洞敌手探测,无法检测自适应改变攻击策略的主动虫洞节点.

- 基于节点邻接性

通过观察网络节点在物理位置上的邻接性可探测虫洞节点,具体而言,虫洞节点间的带外信道可使物理位置上相距较远的节点成为邻居,通过探测节点的邻接性和物理位置的不匹配可发现虫洞节点.文献[4,5]提出了 LiteWorp 和 MobiWorp,在这些方案中,每个节点收集距自己两跳以内的邻居,然后将两跳邻接表发送给某个在网络部署阶段所选择的护卫节点,该节点能通过监听邻接表中非相邻节点的直接通信消息来检测虫洞.但该方案需要在护卫节点上加装定向天线,而且网络中每个节点向护卫节点发送两跳邻接矩阵会造成较大的通信开销,并被虫洞节点感知,因而不能抵御主动虫洞敌手.

为检测可移动的虫洞节点,文献[6]通过比对虫洞节点和普通固定节点邻接表的区别来进行探测.由于虫洞节点在移动中会与不同的普通节点建立连接从而成为邻居,会导致普通节点和虫洞节点的邻接表产生变化,通过对这些所有节点的邻接表进行序列概率比测试(sequential probability ratio test)可发现虫洞节点的移动路径,从而定位虫洞节点.该方法需要每个节点向网络控制端报告其邻接表,会给网络带来较大的通信开销,而且虫洞节点可以控制与移动路径上节点的连接以隐藏自身移动轨迹,从而避免被定位.因此,该方案不能抵抗主动虫洞敌手.

Lu 等人[7]提出让网络中每个节点收集其邻居节点信息,并通过多维标度法构建邻居子图.隐藏的虫洞节点可以通过验证构建的邻居子图的合法性来检测.该方案为分布式探测方法,但只适用于静态网络,即,虫洞节点

均为被动敌手.

- 基于数据包传输时延

该类方法通过观测数据包发送和接收的时间不匹配现象来探测虫洞节点.文献[8−11]中方案假设网络中的控制器和节点能够进行精确的时钟同步来计算往返时延(round trip time,简称 RTT).对于每个数据包,节点都能够估计从源节点到目的节点的大致距离,然后,通过比较距离信息和节点的通信范围来判断是否存在虫洞节点.然而,无线传感网中的精确时钟同步需要网络中所有节点在网络中广播同步数据包,这将带来较大的通信开销;同时,也会作为探测行为的特征被虫洞节点感知.

文献[12]中方案提出在网络中利用移动的信标节点(mobile beacon)向静态的信标节点发送信标包,如果网络中存在虫洞节点,则信标包传输时的通信性质(如转发时延)将会发生改变.检测到虫洞节点后,移动信标节点确定虫洞节点位于其通信范围内,并通过信标节点移动轨迹和其连续的通信范围圆上的弦中垂线交点来估计虫洞节点位置.这些方法需要引入额外的移动信标和静态信标节点,而且检测成功概率受限于移动信标的运动轨迹,且与网络部署的地理大范围大小成反比.另外,该方法并不适用于主动虫洞节点探测.

SLAW 方案[13]的核心思想是:根据相邻定位器之间的异常信息交换时延状况为每个定位器建立一个所谓的冲突集,然后,根据冲突集从有效的定位器中区分可疑的定位器,最后,通过可疑的定位器提出安全定位算法来定位虫洞节点.然而,该方法需要引入专门的定位器节点;此外,传感器节点需要定期和定位器进行信息交换以确定自己的位置,这将引入大量的通信开销.

- 基于网络拓扑

文献[14−16]假设无线传感器网络具有某些特殊的拓扑模型,通过观测网络拓扑不匹配来发现虫洞节点.如:文献[16]提出移动传感器网络中基于邻居拓扑的虫洞探测方法 SWAN,SWAN 利用网络中移动节点收集的本地静态邻居信息来探测虫洞节点,不需要任何的特殊硬件设备且不会带来显著的通信开销;文献[17]根据虫洞隧道引起的网络拓扑异常来探测传统虫洞攻击和拜占庭式虫洞攻击.

总之,在基于网络的拓扑探测中,为了得到拓扑不匹配的特征,网络中的节点需要向外发送专门的探测包,但探测包作为探测过程的标识会被虫洞节点感知,因而无法抵御主动虫洞敌手.

- 基于路径统计

文献[18−21]中方案提出了一种基于路径统计信息的方法,但是这类方法仅仅适用于多路径按需路由协议,不适用于目前无线传感器网络最常采用的最短路径路由机制.

Chen 等人[22]关注于在传感器网络 DV-Hop 定位过程中防御虫洞攻击,以消除虫洞攻击对 DV-Hop 定位的影响.为此,该文献提出了一种基于标签(label)的耐虫洞攻击安全定位策略,主要思想是:首先,根据节点(包括传感器节点和信标节点)的通信性能为它们标注不同的标签;之后,节点能够识别各自的伪邻居(伪邻居表示原来不是邻居节点,由于虫洞攻击才成为邻居节点);接着,伪邻居之间的通信被禁止以进行安全定位.然而,该方法只使用传感器网络 DV-hop 定位应用,无法适用其他应用中的主动虫洞敌手抵御.

Ji 等人[23]首先通过实验分析虫洞攻击对网络编码技术性能所造成的负面影响,接着提出一种集中式的虫洞发现算法,之后提出分布式系统中名为 DAWN 的分布式算法来解决无线网络编码中的虫洞现象.DAWN 通过观察虫洞引起的更新包流方向的变化来探测虫洞攻击.

综上所述,现有工作或者需要额外硬件,或者带来较大的通信和计算开销.更严重的是,这些方法只能探测被动式虫洞,即,这些方法均假设虫洞节点在部署以后只能一直发动攻击——截获并转发数据包,而不能主动感知探测事件的发生从而规避探测.因此,上述方案均不能探测主动虫洞节点.另外,上述方案中,虫洞探测过程具有周期性,即,当网络控制端主动发起探测时才能探测虫洞节点,不能对虫洞进行实时检测.

2 系统模型和设计目标

本节中,我们将首先描述系统模型,包括网络模型和敌手模型,再提出虫洞探测方案的设计目标.

2.1 系统模型

2.1.1 网络模型

无线传感器网络可看做数据收集的一种方式.在网络中,每个节点周期性的采集数据并通过多跳传输将数据发送回 Sink 节点.我们的方案针对不同通信拓扑的网络,例如生成树和有向循环图.

传感器网络的生命周期由两部分组成:系统初始化阶段和运行阶段.在系统初始化阶段,合法用户部署所有网络节点.显然,合法用户不会故意部署虫洞节点,所以在这个阶段没有虫洞节点存在.而随着系统初始化完成并进入运行阶段,网络被敌手入侵后可能会出现虫洞节点.

2.1.2 敌手模型

虫洞敌手可分为两类:被动敌手和主动敌手.被动敌手只吸引流量而忽视网络中的虫洞探测行为.而主动敌手不仅能够吸引网络流量,还能根据网络的状态自适应调整攻击策略.例如,在一些方案中,Sink 需要广播一个控制包以获得全网每个节点的邻接表以探测虫洞节点.而这个控制包可被看作网络启动探测程序的标志,当虫洞节点发现这个包之后,它就可以临时中断攻击行为以躲避探测.

2.2 方案设计目标

在当前的无线传感器网络中,大部分节点的计算能力和通信能力都有限.因此,一个有效的虫洞探测和定位方案必须在这两方面是轻量级的.除此之外,还需满足:

1) 实时探测.虫洞检测方案必须能在网络运行阶段持续检测和定位虫洞.而为了尽可能快地探测虫洞,要求探测延迟尽可能低;
2) 能检测主动虫洞敌手.探测过程中不能出现与网络正常运行不同的事件或者行为,以防止主动虫洞节点感知探测事件的发生.

3 方案详细介绍

在本节中,我们将首先介绍 Pworm 的系统架构,再详细介绍构架中各部分.

3.1 系统架构

如图 2 所示,Pworm 由两大功能模块组成:安全拓扑收集和虫洞探测及定位模块.其中,

- 安全拓扑收集模块的功能是实时收集网络拓扑,主要由数据包标记和拓扑重构两种算法组成.数据包标记算法是指网络中的普通节点对数据包做的特定的标记操作;而网络拓扑重构算法是 Sink 节点对数据包里的标记进行抽取并据此重构网络拓扑.由于网络拓扑重构算法运行在 Sink 端,而通常认为 Sink 端具有较强的计算和通信能力,因此,仅 Sink 端的实时拓扑重构算法的计算开销可不计入 Pworm 的整体开销.与之不同的是,数据包标记算法运行在网络中的普通节点上,需考虑包标记算法的计算和通信开销以节省节点能量;
- 虫洞攻击探测和定位模块的功能是依据实时获取的网络拓扑来检测网络中是否有虫洞攻击行为的发生,并定位虫洞节点在网络拓扑中的位置.该部分也由两个算法组成:虫洞探测和虫洞定位算法.虫洞探测算法的功能是根据网络拓扑的实时变化判断网络中是否有虫洞攻击发生;而虫洞定位算法是在判定了虫洞攻击存在之后对虫洞节点定位,并生成攻击报告.

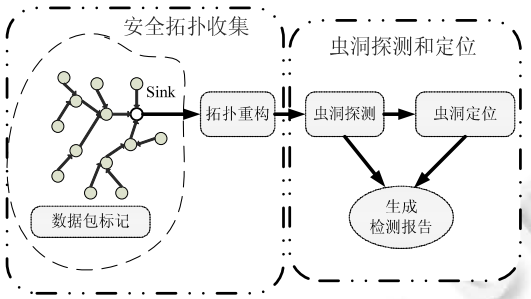


Fig.2 System architecture

图 2 系统架构图

3.2 系统描述

3.2.1 安全拓扑收集

由于传感器网络拓扑的时变性,对于高动态性的自组织网络来说,需要持续、实时地获得传感器网络的拓扑状态;同时,由于存在敌手的恶意攻击行为,可能对数据包中的信息进行篡改,因此,拓扑收集要能抵抗信息篡改.为了满足上述需求,我们设计了轻量级、实时的安全包标记算法和拓扑重构算法.

3.2.1.1 安全数据包标记

在算法中所用到的符号见表 1.

Table 1 Notations

表 1 符号定义

符号	含义
p	数据包
Src_ID	数据包源节点 ID
Sq_N	数据包序列号
Hops	数据包转发时经过的跳数
Mark_ID	对数据包进行标记的节点 ID
Pre_ID	前一节点 ID
MAC	消息验证码字段
K_i	节点 *i* 与 Sink 的共享密钥
MAC_K	用密钥 *K* 计算的消息验证码值
ItrN_i	中间节点 *i*

安全包标记算法的基本思路为:让网络中的传感器节点将自己的 ID 号和距离数据包源节点的距离标记到由它转发的数据包中.但是,由于受到数据包长度的限制(Zigbee 标准中默认数据包长度为 29 字节,经过设置后,数据包长度可以达到 100 字节以上),而且从通信额外负荷的角度来看,也不能让数据包途经的所有节点将自身 ID 标记到数据包中.因此在数据包的转发中,只有一个被选择的节点将自己的 ID 号标记进数据包中,其他节点则会按照一定的规则更新数据包中的跳数域和其他一些必要信息,从而保证拓扑图的正确收集.在 Pworm 中,我们使用两字节来标记节点的 ID 号,使得 Pworm 可以支持节点数不大于 65 536 的传感网络.

在传感器网络中,每个数据包中除了数据部分还包含了以下 3 个部分:(1) 产生数据包的源节点 ID——*Src_ID*;(2) 用于识别数据包的产生时间顺序的序列号——*Sq_N*;(3) 数据包被转发的次数——*Hops*(在本文中,网络拓扑中的距离以跳数作为度量单位).另外,为了标记数据包,Pworm 为数据包设计了一种特殊的数据结构,称为标记结构(如图 3 所示).该结构包含了 3 个部分:(1) 对数据包进行标记的节点 ID 字段——*Mark_ID*;(2) 记录标记节点的前一跳节点 ID 的字段——*Pre_ID*;(3) 用于验证数据包内容完整性的消息认证码字段——*MAC*.

除此之外,在每一个节点上需维护一个数据包缓存 *Cache*,其结构如图 4 所示.*Cache* 记录了由它所转发的最新数据包的序列号、该数据包对应源节点的信息和源节点距离本节点的跳数信息.需说明的是:序列号字段保

存的是该节点所收到的对应源节点所发送的最新数据包的序列号,并且网络中的每一个节点分别都维护一个 *Cache*.

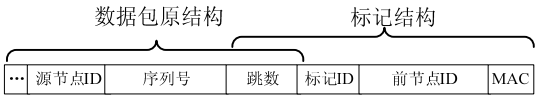


Fig.3 Packet marking structure

图 3 数据包标记结构

……	……	……
源节点ID	序列号	跳数
……	……	……

Fig.4 Node cache

图 4 节点 Cache

当源节点(*Scr*)发送一个数据包 *p* 前,在该数据包的标记结构中,设置 *p.Hops*=1,*p.Mark_ID=NULL*(留空),再将 *Scr* 的 ID 填入 *p.Scr_ID* 和 *p.Pre_ID* 两个字段中,并计算 $p.MAC=MAC_{k\_scr}(Data,Pre\_ID,Sq\_N)$,其中:*Data* 为数据包所携带的数据的内容,生成 *MAC* 的秘钥为 Sink 节点和该节点之间共享的秘钥(下同).数据包在转发过程中需要采用安全包标记算法对被转发的所有数据包进行标记.在标记过程中,每一个中间节点 *ItrN_i* 都维护着一个 *Cache*(如图 4 所示),每条 *Cache* 包含了"源节点 ID"*Src_ID*,由该源节点发出的数据包的最新序列号 *Sq_N* 和中间节点距离该源节点的跳数 *Hops*.

算法 1 所示的安全包标记算法,每当中间节点 *ItrN_i* 收到一个包 *p* 时,中间节点会首先检查 *p* 是否被标记.

- 如果已被标记,即 *p.Mark_ID*≠*NULL*,则 *ItrN_i* 仅将 *p.Hops* 加 1 后直接转发;
- 如果 *p* 未被标记,节点首先会在自己的 *Cache* 中检查是否存在 *p.Scr_ID*:
  - 如果没有,则表示收到了一个来自新源节点的数据包,因此,*ItrN_i* 会为 *p* 在 *Cache* 中创建一个新条目,再标记该数据包:将 *p.Mark_ID=ItrN_i.ID*,并计算新的:
    $$p.MAC=MAC_{k\_ItrN\_i}(p.Mark\_ID,p.MAC,p.Hops);$$
  - 如果中间节点 *ItrN_i* 的 *Cache* 条目里已经储存了 *p.Scr_ID*,并且 *p.Sq_N*=*Cache.Sq_N*+1(即,序列号连续),则 *ItrN_i* 检查 *p.Hops* 是否等于 *Cache.Hops*:如果相等,则 *Cache.Sq_N*=*p.Sq_N*,并设置 *p.Hops*=*p.Hops*+1,同时,*ItrN_i* 设置 *p.Pre_ID=ItrN_i.ID* 和 $p.MAC=MAC_{k\_ItrN\_i}(Data,p.Pre\_ID,p.Sq\_N)$;如果 *p.Hops*≠*Cache.Hops*,则意味着数据包里的跳数信息已被敌手恶意篡改.这是因为 *p.Sq_N* 连续,说明了来自同一源节点的数据包的转发路径没有发生变化;而 *p.Hops* 和 *Cache.Hops* 不同,则意味着路径已发生变化,因此,出现此矛盾的唯一可能是敌手为了阻止网络收集到正确拓扑而恶意更改了数据包里的跳数 *p.Hops* 字段.此时,*ItrN_i* 设置 *p.Hops*=*Cache.Hops* 和 *p.Pre_ID=ItrN_i.ID*.然后,在 *p* 中增加标记 Attack Alert(攻击警告),并且更新 $p.MAC=MAC_{K\_ItrN\_i}(p.Pre\_ID,p.Sq\_N,p.Hops)$.如果 *ItrN_i* 节点 *p.Sq_N*≠*Cache.Sq_N*+1(序列号不连续),可能是因为转发路径更新或者其他原因,在这种情况下,我们就需要对数据包进行标记,其标记过程如前所述.

算法 1 所示的安全包标记算法中,每个数据包仅被一个中间节点标记,同一源节点发送到 Sink 的多个数据包会被路径上的中间节点依次标记,即,路径上第 *i* 个节点仅标记该源节点想 Sink 发送的第 *i* 个数据包,从而节省了通信开销.另外,包标记结构中各字段——*Hops* 字段仅 1 个字节,*Mark_ID* 和 *Pre_ID* 字段均为 2 字节,由此给每个数据包增加的额外通信开销仅为 5 字节.并且,在数据包在转发过程中出现故障,例如丢包或路径改变,包标记算法都会对此做出响应,通过数据包标记的信息反应给 Sink 节点,以及时更新传感器网络的拓扑图,实现了网络拓扑图随数据包变化的快速反应.

算法 1. 安全数据包标记算法.

输入:数据包 *p*,中间节点 *ItrN_i*;

输出:已标记的数据包 *p*.

1. IF 数据包 *p* 已经被标记
2. RETURN;

3. ELSE
4. *ItrN_i* 检查缓存 *Cache*;
5. IF 没有数据包的“源节点”的记录
6. 标记 *p.Mark_ID=ItrN_i*.ID,计算 *p.MAC*=(*ItrN_i.ID*,*p.MAC*,*Hops*);
7. 在 *Cache* 中创建记录:(*p.Scr_ID*,*p.Sq_N*,*p.Hops*);
8. ELSE IF 记录存在而且 *p.Sq_N*=*Cache.Sq_N*+1 //即序列号连续
9. IF (*p.Hops*==*Cache.Hops*)
10. *Cache.Sq_N*=*p.Hops*; //更新缓存记录新的序列号;
11. *p.Hops*+=1,*p.Pre_ID*=*ItrN_i.ID*,*p.MAC*=(*Data*,*p.Pre_ID*,*p.Hops*);
12. ELSE
13. *p.Hops*=*Cache.Hops*;*p.Pre_ID*=*ItrN_i.ID*,*p.MAC*=$MAC_{K\_ItrN\_i}$(*p.Pre_ID*,*p.Sq_N*,*p.Hops*), 将“Attack Alert”加入数据包 *p*;
14. END IF;
15. END IF;
16. ELSE IF 记录存在且 *p.Sq_N*≠*Cache.Sq_N*+1
17. 标记 *p.Mark_ID=ItrN_i.ID*,计算 *p.MAC*=(*ItrN_i.ID*,*p.MAC*,*Hops*);
18. 更新 *Cache.Sq_N*=*p.Sq_N*,*Cache.Hops*=*p.Hops*;
19. END IF;

3.2.1.2 网络拓扑重构

在上一节中,我们详细介绍了 Pworm 中节点对数据包进行标记的过程.当 Sink 收集到一定数量的数据包后,标记抽取模块会对收到的数据包里的标记内容进行提取和验证,重构网络拓扑.为重构拓扑或网络中路由,Sink 为每一个源节点都保存了一个条目,其数据结构如图 5 所示.其中:源节点 ID——*Scr_ID* 表示某条路起始节点 ID;*PathID* 是从 *Scr_ID* 到 Sink 节点所经过的中间节点 ID,数组中的第 *i* 个节点即是路径中第 *i* 个中间节点(距源节点距离为 *i* 跳);除此之外,该数据结构中还定义了一个“数据包序列号”字段 *Sq_N* 记录了从该源节点所发送的最新数据包的序列号.

…	…	…
源节点 ID 号	路径 PathID	数据包序列号
…	…	…

Fig.5 Sink entry structure

图 5 Sink 条目结构

算法 2 中,Sink 每收到一个数据包,*p* 会首先检查是否存在来自于 *p.Scr_ID* 源节点的路径.如果 *p* 是该源节点发送到 Sink 的第 1 个数据包,则 Sink 节点的缓存里必然不存在这条路径.因此,Sink 在缓存中为该源节点创建一条新的路径数组 *Path_ID*(假设该源节点号为 ID).之后,Sink 将源节点 ID 放到 *Path_ID*[0]中,记为 *Path_ID*[0]=*Scr_ID*,表示距离源节点为 0 跳的节点号为 ID.对于数据包 *p*,如果未被标记(*p.Mark_ID=NULL*),这就意味着 *p* 沿着之前的路径传到了 Sink,因此,Sink 只需要更新缓存中对应条目的数据包序列号.

算法 2. 网络拓扑重构算法.

输入:已标记的数据包 *p*;

输出:网络拓扑.

1. IF 当前关于源节点 *p.Scr_ID* 对应的路径
2. 为该源节点创建一个路径信息数据单元,记作 *Path_ID*;

```
3.  ELSE
4.    IF 数据包 p 没有被标记
5.       将 p.Scr_ID 对应路径的 Sink.Sq_N 修改为 p.Sq_N;
6.    ELSE
7.      IF p 中包含“Attack Alert”
8.         为路径 Sink.Path_ID 生成攻击报告;
9.         IF p.MAC 正确;
10.           将攻击位置定位 p.Pre_ID;
11.    ELSE IF p.MAC 不正确
12.             为路径 Sink.Path_ID 生成攻击报告;
13.       ELSE
14.             |d=p.Sq_N−Sink.Sq_N|;
15.              IF d≤1
16.                 IF Sink.Path_ID[p.Hops−1]≠p.Pre_ID
17.                    为 Sink.Path_ID 生成攻击报告,并定位虫洞到 Sink.Path_ID[p.Hops−1]节点;
18.                ELSE IF Sink.Path_ID[p.Hops]==p.Mark_ID
19.                        生成“路径变化”报告;
20.                  END IF;
21.                END IF;
22.              ELSE
23.                生成“丢包”报告;
24.                IF Sink.Path_ID[p.Hops]≠p.Mark_ID
25.                  清空 Sink.Path_ID[p.Hops−d]后所有的节点 ID;
26.                    Sink.Path_ID[p.Hops]=p.Mark_ID;
27.                    生成“路径变化”报告;
```

如果 p 被标记过(即 $p.Mark\_ID$ 不为空),Sink 首先检查数据包里的消息是否是 Attack Alert(攻击警告).

- 如果是 Sink,首先生成攻击报告;接着,Sink 节点会检查 $p.MAC$ 是否正确,即:

$$p.MAC=MAC_{K\_Mark\_ID}(p.Pre\_ID,p.Sq\_N,p.Hops).$$

  如果相等,Sink 将错误定位到 $p.Pre\_ID$ 节点;如果不等,则不对错误位置进行定位;

- 如果 p 中不包含“Attack Alert”,Sink 首先检查包里的 MAC 值是否正确.此时,Sink 需计算两次 MAC 值:
  - 首先计算数据的 $MAC_{Data}=MAC_{K\_Pre\_ID}(Data,p.Pre\_ID,p.Sq\_N)$;
  - 再计算标记结构中的 $p.MAC$ 是否等于 $MAC_{K\_Mark\_ID}(p.Mark\_ID,MAC_{Data},p.Hops)$:如果不等,则说明了网络中有敌手篡改了数据包内容,Sink 节点此时就会生成关于该源节点对应路径的攻击报告;如果相等,Sink 继续重构拓扑.

设 d 为数据包中的序列号和保存在路径信息里的数据包序列号的差值,即 $d=|p.Sq\_N−Sink.Sq\_N|$.在理想情况下,如果数据包都能顺利地被发送和接收,d=1.此时,根据包标记的算法,p 被标记的唯一原因是网络路由发生了变化.接着,Sink 检查 $p.Pre\_ID$ 是否等于 $Sink.PathID[Sink.Hops−1]$,即,数据包中前跳节点 ID 是否等于 Sink 中该路径中相同位置的节点 ID:如果相等,Sink 生成路由变化报告,然后将条目中对应位置的 ID 替换为 $p.Mark\_ID$;如果不相等,则意味着数据包必然发生了路径变化的情况,而一个被标记过的数据包只能说明路径中的一个特定的位置发生了变化,且这个特定的位置必定为标记节点的位置,所以此时如果标记节点的前跳节点与路径中的对应位置不相等,则必然出现了攻击行为.此时,Sink 节点将生成一个攻击报告,并将攻击者定位到该路径 $Path\_ID$ 中的标记当前数据包标记节点所

处位置的上一跳位置.

如果两个序列号不连续,即 $d>1$,则意味着传感器网络中必定发生了丢包,Sink 会首先产生一个丢包的报告.之后,Sink 检查数据包的 *p.Mark_ID* 是否等于 *Sink.Path_ID*[*p.Hops*],即,*p* 中记录的标记节点是否与 Sink 的记录一致:如不同,说明路径发生变化,Sink 将清楚对应 *p.Scr_ID* 条目中,在 *Path_ID*[*p.Hops*−*d*]之后的所有中间节点(由于路径发生变化,标记节点之后的节点不再属于原路径),并将新路径中的相应位置储存为数据包中的标记节点 *p.Mark_ID*.最后,Sink 再生成路径变化报告.

数据包标记抽取模块根据记录的所有 *Path_ID* 路径信息实时构建和更新整个网络拓扑图.一旦 Sink 节点收到一个新的数据包,那么与该数据包相关的路径信息就会被更新.在 Sink 节点端,如果出现了数据包中的转发节点和 *Path_ID* 路径信息中的对应节点不匹配的情况,这就意味着网络中发生了路由改变,也就代表了网络的拓扑图已经出现了变化.此时,将新的转发节点保存到 *Path_ID* 的信息中去以替换原有信息.从而为进行更加深入的网络分析进行了数据统计.

从标记算法可以看出:仅仅通过一个数据包不能构建出全网拓扑,除非源节点离 Sink 只有一跳的距离.所以,我们需要根据一定量的数据包来重构出数据包传输路径.显然,所需数据包的数量至少应等于源节点到 Sink 的跳数.目前,传感器网络的路径长度一般都低于 10 跳,因此,重构路径所需的数据包数量也较少.除此之外,在传感器网络中由于存在大量的路径共享现象,拓扑重构算法甚至能在不知道所有的路径信息的情况下,通过信息互补重构出完整的网络拓扑.

3.2.2 虫洞探测

探测算法的目的是,通过分析实时网络拓扑来判定是否有虫洞攻击的发生.虫洞节点通过一条高速带外信道来减少数据包传输的时间和距离,以够吸引网络中的流量.由此可知,虫洞攻击至少具有两个基本特征:(1) 网络中路径长度缩短;(2) 网络中大量流量聚集.依据这两个特征,可对虫洞攻击进行探测.探测算法的具体步骤见算法 3.

算法 3. 虫洞探测算法.

输入:网络初始拓扑图记为 U,网络实时拓扑图记为 U_{RT};

输出:对虫洞节点存在性判定.

1. 比较路径集合 U 和 U_{RT},得到 U 和 U_{RT} 发生变化的路径集合分别为 $\bar{U}$ 和 $\bar{U}_{RT}$:

$$\bar{U}=(r_{k_i},r_{k_2},...,r_{k_t}),$$

$$\bar{U}_{RT}=(s_{k_i},s_{k_2},...,s_{k_t}).$$

r_{k_i} 和 s_{k_i} 分别表示原始路径和变化后的路径;

2. IF $t/n \geqslant Threshold_R$

3. IF $\sum_{i=1}^{t}(|r_{k_i}|-|s_{k_i}|)/t \geqslant Threshold_L$

4. 网络中存在虫洞攻击;

5. ELSE

6. 网络中没有虫洞攻击;

7. END IF;

8. END IF;

如第 2.1 节所述,传感器网络由合法用户部署,因此,我们假设合法用户在网络部署时不会加入虫洞节点,当网络运行一段时间之后,虫洞节点才可能被敌手部署到网络.因此,在网络部署之后、虫洞节点被部署之前,存在一个“干净”的初始网络拓扑(记为 U),该拓扑可使用如前所述的安全拓扑收集方法获得.U 由网络中的所有节点到 Sink 节点的路径组成,记为 $U=(r_1,r_2,\ldots,r_n)$.r_i 表示从节点 i 出发到 Sink 的数据包所走过的路径,设网络中路径总数为 n.在网络运行阶段,系统将实时的收集到网络的拓扑信息,记为 $\bar{U}_{RT}=(s_{k_i},s_{k_2},...,s_{k_t})$,其中,$s_i$ 表示从节点 i 出发到 Sink 的实时路径.由于网络中可能出现路径变化,故 r_i 和 s_i 可能会不同.

当网络持续运行时,Pworm 周期性地将实时的网络拓扑图 $\bar{U}_{RT}$ 与初始网络拓扑图 U 比较,找出其中变化的路径.记 U 中变化路径为 $\bar{U}=(r_{k_1},r_{k_2},...,r_{k_t})$,$\bar{U}_{RT}$ 中的变化部分为 $\bar{U}_{RT}=(s_{k_1},s_{k_2},...,s_{k_t})$,$t$ 表示变化路径的条数.由于虫洞节点会吸引大量流量,因此在算法 3 中,我们用变化路径占网络总路径的比例(t/n,n 为网络中路径总数)来衡量虫洞节点对流量的吸引效果.但是由于传感网络中无线传输的不可靠性,因此网络动态性(即,网络路径发生变化),这就容易与虫洞链路引起的网络动态性相混淆.但是在同一时刻,网络动态性引起的路径变化并不会导致网络内大量的路径变化.因此在算法 3 中,我们定义比例阈值——$Threshold_R$ 对这种吸引效果进行判断:如果 $t/n<Threshold_R$,我们认为路径变化是网络动态性引起的,因为它对路径变化的影响不明显;如果 $t/n>Threshold_R$,则认为可能存在虫洞节点.

事实上,如果虫洞节点部署位置不好,例如,两个虫洞节点间若距离太近则无法吸引大量流量,那么虫洞攻击对网络路径变化的影响不明显,我们把这种情形下的错误判断作为漏报.如果 t/n 大于阈值 $Threshold$,意味着网络中发生了异常事件,为提高判定准确率,Pworm 将进行二次判定,其依据为路径的缩短长度.

第 2 步判定能将网络动态性和真实虫洞攻击区分开来.由于网络采用最短路径算法,因此在初始网络拓扑 U 中的路径必将分别是网络中的最短的路径,即便出现网络动态性,变化后的网络路径也不会比 U 中的路径更短.但是虫洞节点间的带外信道将大幅缩短网络中的路径长度,因此在虫洞攻击引起的变化路径中,平均路径长度将被大幅缩短.据此,我们定义另一阈值 $Threshold_L$,称为长度阈值,来描述网络平均路径长度减少的跳数.如果网络中平均长度减少量大于 $Threshold_L$,可认为网络中出现了虫洞攻击.严格说来,正常情况下路径并不可能被缩短,因此长度阈值理论上为 1.但是由于无线网络中不确定性因素较多,为了减少误判率,可以将长度阈值定为 1.5 来进一步区别虫洞链路和正常网络动态性所引起的网络拓扑变化.

3.2.3 虫洞定位算法

一旦确定虫洞攻击发生,将使用虫洞定位算法定位虫洞.定位目标是:如果虫洞节点出现在网络拓扑中,则最终定位到虫洞节点本身;如果虫洞节点不出现在实时网络拓扑中,则定位到其邻居节点.由于虫洞节点会导致网络中的路径变化,因此虫洞节点必然只存在于变化的路径中.据此可定位虫洞节点,举例来说,如图 6 所示,节点 1 和节点 5 为虫洞节点,箭头代表虫洞链路,其中,节点 5 为虫洞出口节点,节点 1 为虫洞入口节点.当网络初始阶段,从节点 0 到 Sink 之间的传输路径为 0→1→2→3→4→5→6→Sink;在发生了虫洞攻击之后,从节点 0 到 Sink 的传输路径为 0→1→5→6→Sink.比较这两条路径可发现:虫洞出口节点距离 Sink 节点的距离其实并未发生任何变化,即,虫洞出口节点到 Sink 的距离在拓扑变化后不变;而虫洞入口节点在网络中的拓扑位置却发生了变化,变为 1→5→6→Sink.因此,根据此检项可先定位虫洞入口节点,再将入口节点的下一跳邻居作为出口节点.

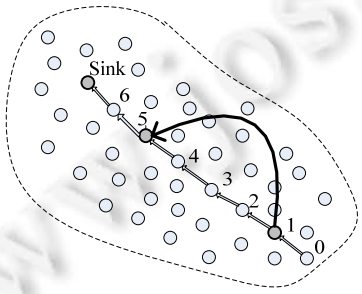


Fig.6 Wormhole positioning schematic

图 6 虫洞定位示意图

综上,虫洞定位算法见算法 4.

算法 4. 虫洞定位算法.

输入:在发生变化的路径中,网络初始路径集合记为 $\bar{U}$,网络实时路径记为 $\bar{U}_{RT}$;

输出:虫洞节点对在网络中的位置.

1. 找出所有 $\bar{U}_{RT}$ 中距 Sink 节点距离变短的节点集合 I;
2. 找出 I 中距离 Sink 最短的节点,该节点即为虫洞节点的入口节点,记为 $Wormhole_{in}$;
3. 找出在集合 $\bar{U}_{RT}$ 中 $Wormhole_1$ 的距 Sink 更近的邻居节点作为虫洞节点的出口节点,记为 $Wormhole_{out}$;

4 安全性分析和网络动态性

在本节中,我们首先分析 Pworm 的安全性,再对网络动态性对 Pworm 有效性的影响进行讨论.

4.1 安全性分析

在 Pworm 中,虫洞敌手可以是被动敌手或主动敌手:被动敌手只能接收数据包并通过虫洞间的高速信道进行转发,而主动敌手可以自适应地调整攻击策略以防止被检测出来.但是由于 Pworm 将网络拓扑收集和虫洞检测过程融入了网络正常运行中,因而探测事件难以被被虫洞节点所感知.由此,即使主动敌手也不能躲避探测.然而,主动敌手可以故意通过错误标记数据包或随机转发数据包的方式破坏正确网络拓扑的收集.因此,在安全性分析中,我们将分别分析 Pworm 在面对被动和主动虫洞敌手时的安全性.

4.1.1 被动敌手

为了躲避探测,被动敌手可选择是否将自己的节点 ID 标记进数据包:如果敌手标记自己的 ID,则 Sink 端可以发现虫洞节点并能定位出虫洞节点所在的位置;如果敌手不标记自身 ID,虽然 Sink 无法在路径信息中发现虫洞节点的 ID,但虫洞节点的邻居节点会表现出虫洞特征,并且邻居节点会在数据包中标记自身 ID,故而 Sink 能定位虫洞节点的邻居节点.

4.1.2 主动敌手

主动敌手可通过篡改数据包标记内容的方式破坏 Sink 对网络拓扑的正确收集以躲避检测.如第 3.2.1.1 节所述,安全包标记算法利用消息认证码防止标记内容被恶意篡改,它保护了标记域中的除了跳数域以外的所有内容.因此,主动敌手只能修改数据包中的跳数域,如果修改其他域他会在标记抽取阶段中被 Sink 通过检测消息认证码发现.如果敌手伪造某个数据包标记中的跳数域,则对该数据包的处理存在如下两种情况:(1) 沿着之前的路径将数据包转发到下一跳;(2) 随机选择另外一个邻居节点转发数据包.

- 对于第 1 种情况,当虫洞节点的邻居收到数据包时,它会在自己的 Cache 中查询相应路径的条目,然后比较 Cache 中的跳数和数据包中的跳数.由于敌手篡改了数据包中的跳数值,所以该邻居节点会发现 Cache 和数据包中的跳数值不一致,从而立即发现并报告有敌手对标记进行了修改;
- 对于第 2 种情况,主动敌手随机选择一个其他邻居进行转发.当某个邻居节点收到该数据包之后,会认为该包来自一个新的源节点,从而会为数据包在其 Cache 中创建新条目.但是,Sink 在路径抽取过程中会发现标记被修改.确切地说,Sink 会检测发现来自同一源节点的数据包路径的不一致,从而发现数据包标记被修改,进而发现和定位攻击.另外,如果敌手随机丢弃数据包以防止被发现,Sink 端的路径抽取模块可以发现网络频繁丢包,进而定位丢包节点并且发出警告.

4.2 网络动态性

Pworm 的对虫洞攻击的探测基于对路径变化比例的观察,但是网络动态性也会改变网络中部分路径.另外,对于某些网络,其通信链路不固定,传输能力方向不对称也会对网络路径变化产生一定影响.例如,网络中某些节点能量耗尽后停止工作,那么节点所在路径就会断裂而路径中的其他节点就会另外选择路径转发数据包.虽然 Pworm 设置了阈值来区别网络动态性和虫洞攻击,但是网络动态性仍然有可能引起误报,即,将网络动态性判定为虫洞攻击.但是不同于虫洞攻击,由如通信链路变化导致的网络动态性不会导致如同虫洞节点流量吸引效应一样的流量大量聚集在少数虫洞节点上,因此也不会引起网络路径的显著减少.另外,即使节点传输能力方向不对称,只要网络中能形成路由,也不会导致流量产生聚集效应,因而和虫洞节点部署后所行程的路径变化在程度上不同.所以,Pworm 采用网络路径变化比例和网络路径缩短长度作为判定依据来推断网络中是否有虫洞发生,能显著提高检测的准确率.在实验中,采用了上述两项判定依据后,Pworm 的误报率为 0,说明网络动态性对

Pworm 的判定有效性不产生显著影响.

5 实验评价

5.1 实验设置

我们通过 NS-2 仿真工具来验证 Pworm 的性能.在实验中,我们在 25m×25m 的方格内随机部署 100 个~800 个不同节点规模的网络,并在网络中随机部署虫洞节点.每个普通节点每秒产生并发送一个数据包,每个数据包大小为 120 字节,网络带宽为 10Kbp,每个传感器节点的默认通信半径为 25m~50m.我们设置了两种类型的虫洞节点:被动敌手和主动敌手.被动敌手只能实现最基本的虫洞攻击功能,即,只能通过高速带外虫洞链路来吸引网络流量;而主动敌手不仅能实现被动敌手的功能,虫洞的出口节点还能随机挑选一个邻居节点转发数据包以达到均衡负载、从而躲避网络对其探测.

5.2 评价指标

- 有效性

对于有效性的评价指标,可使用误报率和漏报率:误报率是指网络中没有发生虫洞攻击而错误地报告攻击发生,如网络动态性引起的误报;漏报率指网络中发生了虫洞攻击而未被检测出来.

- 实时性

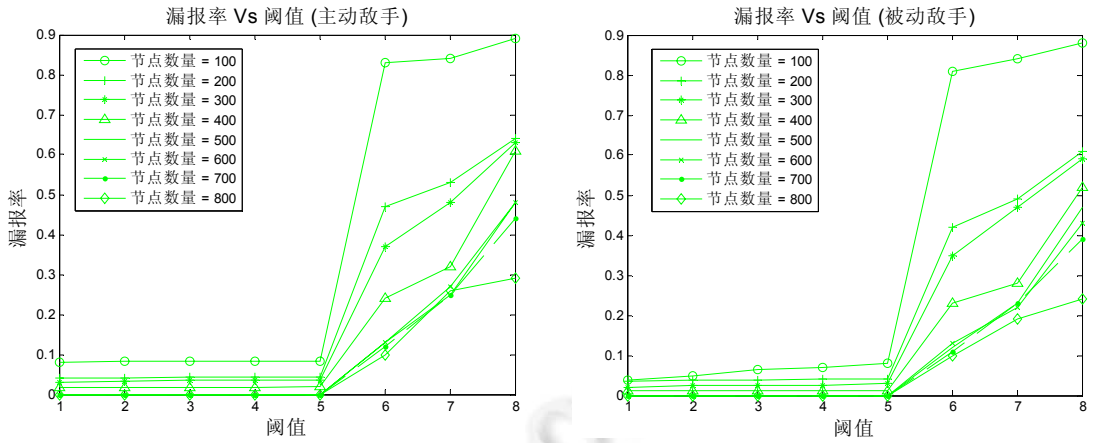
Pworm 时延可分为两部分:激活时延和检测时延.激活时延指的是从虫洞节点部署到网络中时到检测算法被激活进行探测的时延;检测时延是检测算法从激活后到最终定位出虫洞节点的时延.激活时延可评价 Pworm 对虫洞引起的网络拓扑变化的反应实时性;而检测时延则评判的是检测过程的实时性.

5.3 实验结果

需要说明的是:在所有实验中,Pworm 的误报率均为 0.因此,我们在下文中不再对误报率进行分析说明,而从漏报率、时延和可扩展性对 Pworm 进行评价.

5.3.1 漏报率

如图 7 所示,我们以 800 个节点为例展示了在我们的方案在不同阈值下分别对被动敌手和主动敌手的检测漏报率,其中,图 7(a)是为主动敌手,图 7(b)为被动敌手的检测漏报率.从图中可以看到:当阈值低于 5%时,不论是主动敌手还是被动敌手,Pworm 的漏报率均低于 5%;当阈值高于 5%时,方案的漏报率急剧升高.因此,阈值的有效范围为[1%,5%].实验结果表明,可以借助阈值来调整虫洞攻击的探测精确度.较低的阈值表示对探测事件的响应较快,但探测的精确度随着降低;反之亦然.当阈值较低时,虫洞探测的精确度降低.这主要是因为路径的变化被认为是网络动态性造成的.我们同样观察到:当节点的数量增加时,漏报率随之降低,尤其当节点数量从 100~400 变化时特别明显.图 7 还说明:当节点的数量从 100 增加到 200 时,漏报率显著下降,并且这样显著的下降效果持续到节点数量为 400 时为止.当网络中节点的数量位于 500~800 之间时,无论在主动敌手还是被动敌手检测中,随着节点数量的变化,漏报率的变化不大.造成这种现象的原因是:网络的总流量会随着节点的增加而增大,而当网络总流量增加时,虫洞吸引小的网络流量或是改变不够长度的路径的几率随之降低.如 $t \leqslant Length\ threshold$.也就是说,随着网络中节点数量的增加,网络的总流量随之增加,从而进一步导致虫洞保持休眠状态的概率降低.



(a) 主动敌手 (b) 被动敌手

Fig.7 False negative rate vs. threshold for 100~800 nodes shown for active and passive adversaries

图 7 100~800 个节点情况下,主动和被动敌手中漏报率与阈值的关系图

5.3.2 时 延

Pworm 的时延由两部分组成:激活时延和检测时延.图 8 显示了主动和被动敌手攻击下两种时延随阈值的变化情况,其中,图 8(a)为 100 个~400 个节点组成的稀疏网络,图 8(b)则为 500 个~800 个节点下密集网络的延迟变化情况.

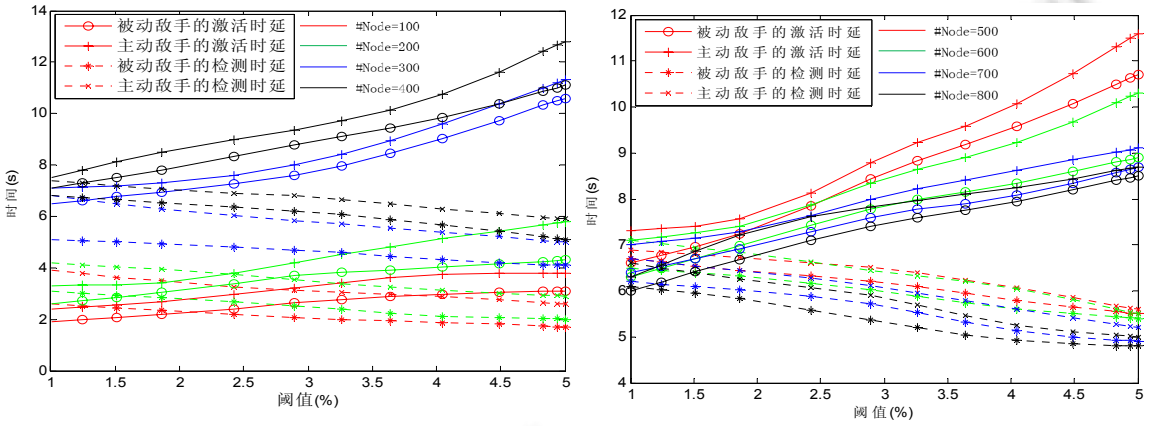


Fig.8 Time delay vs. threshold for different number of nodes shown for active and passive adversaries

图 8 不同节点情况下,主动和被动敌手中时延与阈值的关系图

从图中可以看到:随着阈值的增大,激活时延在增加.这是因为阈值越大,意味着 Sink 节点需要收集更多的变化路径才能激活检测.另一方面,阈值越大,检测时延越短.这是因为 Sink 在收集到足够的变化路径后才启动检测,如果收集到的变化路径越多,则意味着可供判断的信息越多,从而能够更快速地做出判断.此外,从图中可以看到:不论是激活时延还是检测时延均小于 7s,这意味着从虫洞节点进入网络开始最多 14s 的延迟之后,我们的方案就能有效地检测出虫洞节点.

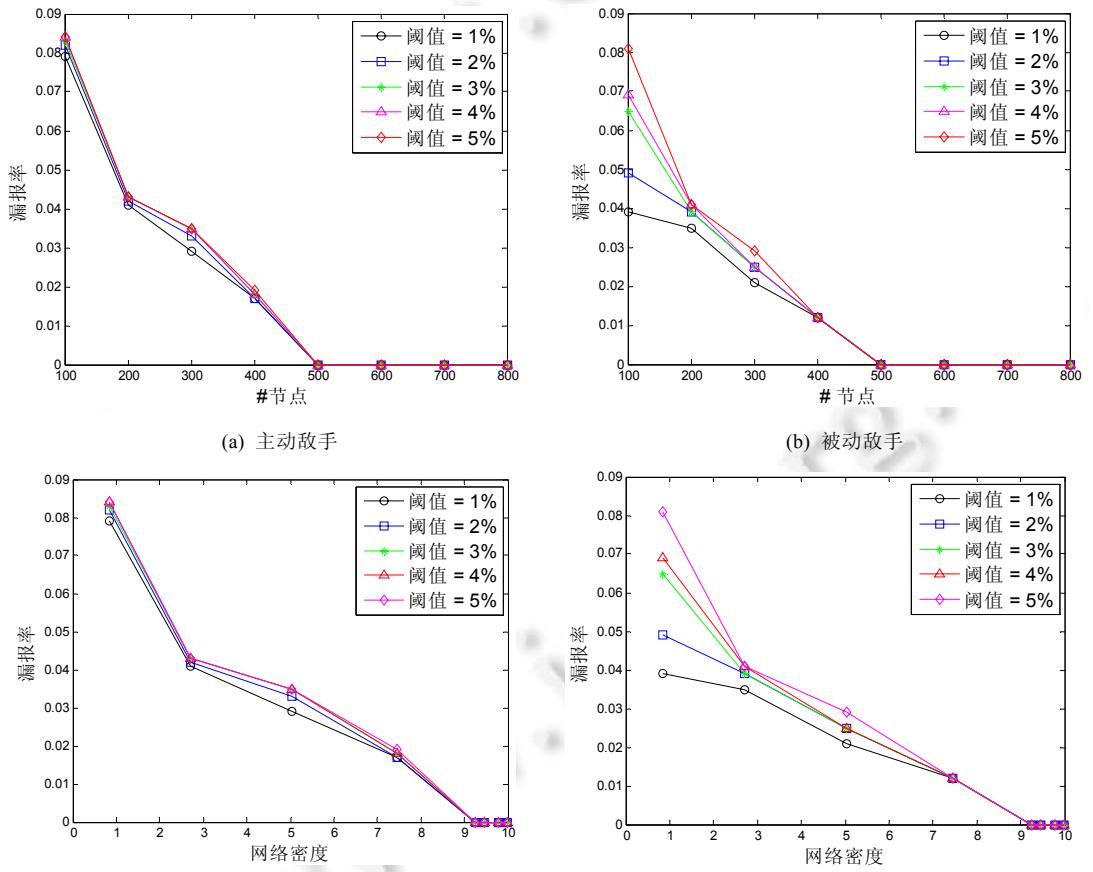
由于图 8(a)中节点数量在 100~200 时网络流量低且收集的路径短,激活时延和检测时延具有几乎相同的取值.但两种时延的变化趋势不同,例如:激活时延随着阈值的增加而增长,但检测时延的变化趋势正好相反.然而,当网络中节点的数量从 300~400 变化时,激活时延大于检测时延.在图 8(b)中,当节点从 500 个增加到 800 个的

过程中,无论节点的数量在该区间中如何变化,两类延迟的值几乎一致.当把主动敌手与被动敌手进行比较我们发现:主动敌手中激活时延和检测时延的变化趋势与被动敌手中两种延迟的变化趋势相同,且被动敌手中的时延始终略低于相同情况下主动敌手中的时延.总之,图 8 中的结果显示:在主动敌手和被动敌手中,Pworm 均能够有效地部署并检测出虫洞节点.

5.3.3 可扩展性

为了验证方案的可扩展性,我们针对不同的网络规模进行了实验,这些网络的节点数量为 100~800.对于漏报率,图 9 展示了阈值在 1%和 5%时的漏报率,其余阈值时其漏报率在上述两者之间.

从图 9(a)和图 9(b)中可以看到:主动敌手和被动敌手的漏报率和节点的数量有关,且漏报率随着节点数量的增加而下降.这是因为网络较稀疏时,虫洞吸引的网络流量有限,因此不能引起足够的网络拓扑变化来触发对虫洞的探测.此外,主动敌手和被动敌手在不同阈值的变化规律大概相同,并且当节点数量较少时漏报率较低,但当节点数量从 100 增大到 400 时,漏报率显著下降.当网络中节点数量在 500 个以上时,网络的漏报率下降到可忽略.



(a) 主动敌手 (b) 被动敌手

(c) 主动敌手 (d) 被动敌手

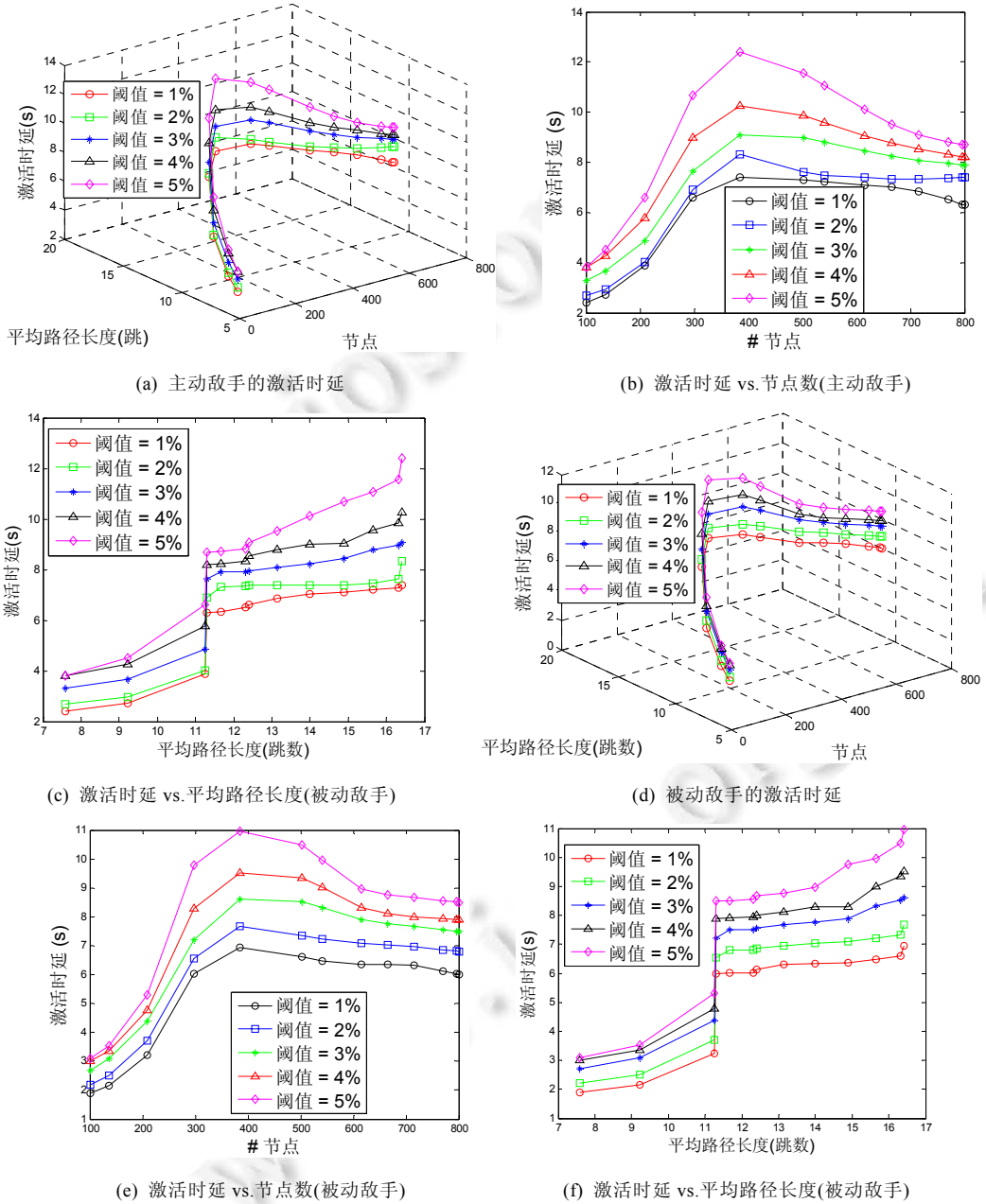
Fig.9 False negative rate vs. nodes for active and passive adversaries, while the false negative rate vs network density for active and passive adversaries

图 9 主动敌手和被动敌手模型中,节点 vs.漏报率以及网络密度 vs.漏报率

图 9(c)和图 9(d)显示了漏报率与网络密度的关系,可见,主动敌手和被动敌手在图中的变化规律基本相同,表现为:随着网络密度提高,漏报率的值显著下降;当网络密度达到 10 时,漏报率的值小到可忽略.引起这种现象

的原因是:节点数量增多时,网络总流量提高并且网络的路径变化增大,这些变化容易被 Sink 节点发现.另外,阈值和网络规模相同时,对于主动敌手的探测漏报率比探测被动敌手的漏报率更低.这是因为主动敌手采取了反探测策略,如随机转发或篡改数据包,反而让 Sink 能更容易判断攻击发生.

图 10 展示了 Pworm 在不同阈值和敌手模型下的激活时延.



(a) 主动敌手的激活时延

(b) 激活时延 vs.节点数(主动敌手)

(c) 激活时延 vs.平均路径长度(被动敌手)

(d) 被动敌手的激活时延

(e) 激活时延 vs.节点数(被动敌手)

(f) 激活时延 vs.平均路径长度(被动敌手)

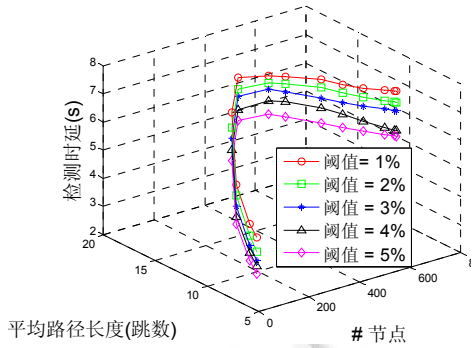
Fig.10 Activation latency shown for active and Passive adversaries

图 10 主动敌手和被动敌手的激活延迟变化

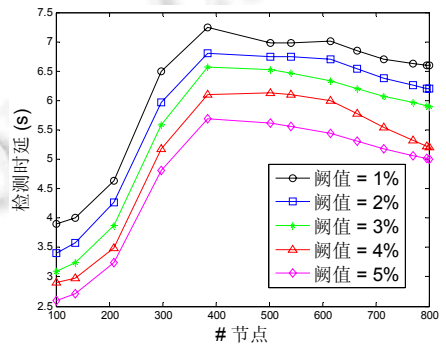
从图 10 中可以看出,激活时延变化趋势为:激活时延先随着网络规模的增大而增长;当网络规模达到 300 个~400 个节点时,激活时延的值最大;之后,激活时延的值逐渐缩短.原因主要来自两方面:首先,当网络规模较少

时,网络路径较短,此时,Sink 节点能很快地收集网络拓扑信息,这时,激活时延也较短;其次,当节点数量大于 300 个节点时,随着节点数量增大,网络中节点更加密集,每个节点几乎都能找到一条更短的路径到达 Sink 节点,使得网络平均路径长度反而变短,此时,激活时延减少.图 10(c)为激活时延随网络平均路径长度的变化关系,可以看出:网络规模小时,激活时延较短.图 10(d)~图 10(f)所示被动敌手模型中,激活时延的变化情况类似于图 10(b),但阈值相同时,被动敌手中激活时延的值比主动敌手中更少.

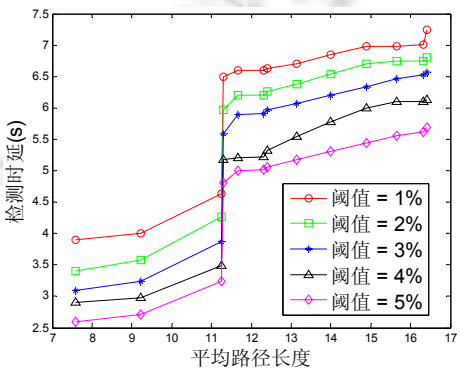
图 11 为不同阈值和敌手模型下的检测时延.检测时延的变化非常类似于激活时延的变化情况,只是检测时延的值略高.



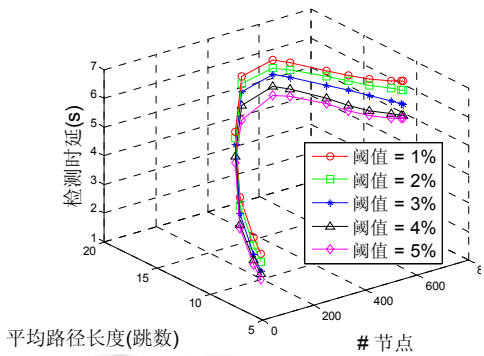
(a) 主动敌手的检测时延



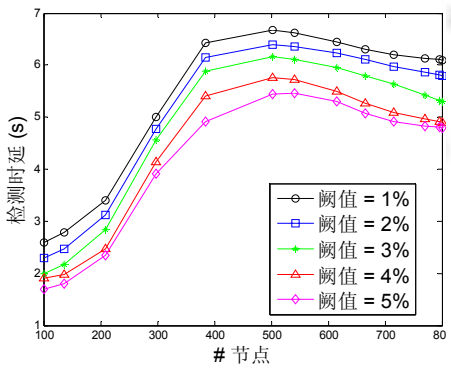
(b) 检测时延 vs.节点数(主动敌手)



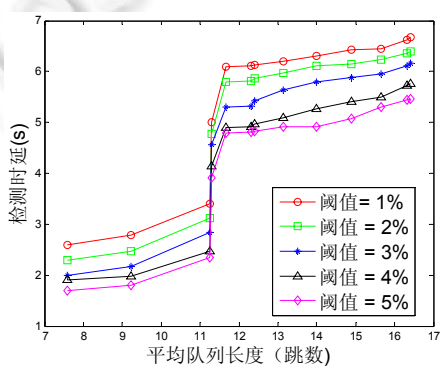
(c) 检测时延 vs.平均路径长度(被动敌手)



(d) 被动敌手的检测时延



(e) 被动敌手的检测时延



(f) 检测时延 vs.平均路径长度(被动敌手)

Fig.11 Detection latency shown for active and passive adversaries

图 11 主动敌手和被动敌手的激活检测变化

我们可以看出:网络中节点数量越多时,网络的总流量和路径的长度越大.此时,Sink 节点收集网络路径所需时间也变长,使得检测时延增大.

总之,由图 10 和图 11 可见:阈值较大时,激活时延和检测时延均增长.这是因为阈值较大时,Sink 节点需要较长时间来观测变化路径的比例.不同阈值时,检测时延和激活时延的变化规律一致,均为网络规模较小时,时延以较快的速度提升到一个最大值;之后,随着网络越来越密集而降低.图 10(c)、图 10(f)以及图 11(c)、图 11(f)中,网络平均路径长度的变化也遵循先增大后减小的趋势.路径长度短时,两种时延的值均较小;反之,网络规模的增大,引起延迟和平均路径长度均增大.

6 结 论

对于无线传感器网络和无线自组织网络而言,虫洞攻击是最严重的威胁之一.现有的虫洞探测方法或者需要较强的网络拓扑假设,或需要额外硬件等,这些问题限制了现有方法在传感器网络中的应用.本文基于虫洞节点对网络流量和拓扑影响的观察,提出了一个轻量级、被动式、实时的虫洞攻击探测方案 Pworm.通过安全包标记,我们将检测过程融入正常的数据包转发过程中,再利用虫洞攻击的两个特征——路径变化和路径长度缩短对虫洞进行实时检测.因此,Pworm 不仅能探测主动和被动虫洞敌手,还能定位虫洞的位置.实验结果表明:我们的方案误报率为 0,漏报率低于 5%,整体时延不超过 14s,能实时、准确的检测虫洞节点.

References:

[1] Capkun D, Buttyán L, Hubaux JP. Sector: Secure tracking of node encounters in multi-hop wireless networks. In: Proc. of the ACM Workshop on Security of Ad Hoc and Sensor Networks. 2003. 21−32. [doi: 10.1145/986858.986862]

[2] Dong DZ, Li M, Liu Y, Li XY. Topological detection on wormholes in wireless ad hoc and sensor networks. In: Proc. of the IEEE ICNC. 2009. 314−323. [doi: 10.1109/ICNP.2009.5339673]

[3] Ban XM, Rik S, Cao J. Local connectivity tests to identify wormholes in wireless networks. In: Proc. of the ACM MobiHoc. 2011. 13−24. [doi: 10.1145/2107502.2107519]

[4] Khalil I, Bagchi S, Shroff NB. Mobiworp: Mitigation of the wormhole attack in mobile multihop wireless networks. In: Proc. of the IEEE Securecomm and Workshops. 2006. 1−12. [doi: 10.1109/SECCOMW.2006.359564]

[5] Hu L, Evans D. Using directional antennas to prevent wormhole attacks. In: Proc. of the NDSS. 2004. 1−9.

[6] Ho JW, Wright M, Das SK. Distributed detection of mobile malicious node attacks in wireless sensor networks. Ad Hoc Networks, 2012,10(3):512−523. [doi: 10.1016/j.adhoc.2011.09.006]

[7] Lu XP, Dong DZ, Liao XK. MDS-Based wormhole detection using local topology in wireless sensor networks. Int'l Journal of Distributed Sensor Networks, 2012,2012:1−9. [doi: 10.1155/2012/145702]

[8] Hu YC, Perrig A, Johnson DB. Packet leashes: A defense against wormhole attacks in wireless networks. In: Proc. of the IEEE INFOCOM. 2003. 1976−1986. [doi: 10.1109/INFCOM.2003.1209219]

[9] Wang X, Wong J. An end-to-end detection of wormhole attack in wireless ad-hoc networks. In: Proc. of the Computer Software and Applications. 2007. 39−48. [doi: 10.1109/COMPSAC.2007.63]

[10] Lazos L, Poovendran R. Serloc: Secure range-independent localization for wireless sensor networks. In: Proc. of the ACM Workshop on Wireless Security. 2004. 1−8. [doi: 10.1145/1023646.1023650]

[11] Wu JF, Chen HL, Lou W, Wang ZB, Wang Z. Label-Based DV-hop localization against wormhole attacks in wireless sensor networks. In: Proc. of the IEEE Networking, Architecture and Storage. 2010. 79−88. [doi: 10.1109/NAS.2010.41]

[12] Chen HL, Chen WD, Wang ZB, Wang Z, Li YJ. Mobile beacon based wormhole attackers detection and positioning in wireless sensor networks. Int'l Journal of Distributed Sensor Networks, 2014, 2014. 1−10. [doi: 10.1155/2014/910242]

[13] Chen HL, Lou W, Wang Z. Secure localization against wormhole attacks using conflicting sets. In: Proc. of the 29th IEEE Int'l Performance Computing and Communications (IPCCC 2010). 2010. 25−33. [doi: 10.1109/PCCC.2010.5682340]

[14] Wang W, Bhargava B. Visualization of wormholes in sensor networks. In: Proc. of the ACM Workshop on Wireless Security. 2004. 51−60. [doi: 10.1145/1023646.1023657]

[15] Xu Y, Ouyang Y, Le Z, Ford J, Makedon F. Analysis of range-free anchor-free localization in a WSN under wormhole attack. In: Proc. of the ACM MSWiM. 2007. 344−351. [doi: 10.1145/1298126.1298185]

[16] Song SJ, Wu HJ, Choi BY. Statistical wormhole detection for mobile sensor networks. In: Proc. of the IEEE ICUFN. 2012. 322−327. [doi: 10.1109/ICUFN.2012.6261721]

[17] Chan KS, Alam MR. Topology comparison-based wormhole detection for MANET. Int'l Journal of Comunication Systems, 2014, 27(7):1051−1068. [doi: 10.1002/dac.2397]

[18] Maheshwari R, Gao J, Das SR. Detecting wormhole attacks in wireless networks using connectivity information. In: Proc. of the INFOCOM. 2007. 107−115. [doi: 10.1109/INFCOM.2007.21]

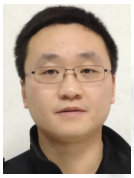
[19] Song N, Qian L, Li X. Wormhole attacks detection in wireless ad hoc networks: A statistical analysis approach. In: Proc. of the IEEE IPDSP. 2005. 8−15. [doi: 10.1109/IPDPS.2005.471]

[20] Zhao Z, Bo W, Dong X, Yao L, Gao F. Detecting wormhole attacks in wireless sensor networks with statistical analysis. In: Proc. of the IEEE ICIE. 2010. 251−254. [doi: 10.1109/ICIE.2010.66]

[21] Buttyán L, Dóra L, Vajda I. Statistical wormhole detection in sensor networks. Security and Privacy in Ad-hoc and Sensor Networks, 2005,3813:128−141. [doi: 10.1007/11601494_11]

[22] Chen HL, Lou W, Wang Z, Wu JF, Wang ZB, Xia AH. Securing DV-hop localization against wormhole attacks in wireless sensor networks. Pervasive and Mobile Computing, 2015,16:22−35. [doi: 10.1016/j.pmcj.2014.01.007]

[23] Ji SY, Chen TT, Zhong S. Wormhole attack detection algorithms in wireless network coding systems. IEEE Trans. on Mobile Computing, 2015,14(3):660−674. [doi: 10.1109/TMC.2014.2324572]



鲁力(1978－),男,贵州铜仁人,博士,副教授,博士生导师,CCF 会员,主要研究领域为无线系统安全.

Muhammad Jawad HUSSAIN (1982－),男,博士生,主要研究领域为无线系统安全.

朱金奇(1981－),女,博士,副教授,主要研究领域为无线网络及安全.