文章编号:1007-5321(2018)03-0032-07 **DOI**:10.13190/j.jbupt.2017-228

WSN 异步休眠模式下节点捕获早期检测方法

张志华[1,2], 罗守山[1,2], 朱洪亮[1,2], 辛 阳[1,2,3]

(1. 北京邮电大学 网络空间安全学院, 北京 100876; 2. 北京邮电大学 灾备技术国家工程实验室, 北京 100876;
3. 北京安码科技有限公司 网络安全研究院, 北京 100082)

摘要: 针对异步休眠模式的无线传感器网络(WSN),提出了一种节点捕获早期检测方法,在被捕获节点重新加入网络之前即被检出,实现对网络攻击的及早发现. 该方法基于相邻节点间存活性监控,并通过节点声明消息统一调度广播机制,保证异步休眠节点间声明消息的接收,同时,在决策过程中采用本地协同决策方式以提高正确率. 仿真结果表明,该方法在检测率、误报率和漏报率等有效性方面均优于其他现有典型方法.

关 键 词: 无线传感器网络; 物理捕获; 节点妥协; 异步休眠; 早期检测

中图分类号: TP393 **文献标志码**: A

A Node Capture Early Detection Scheme for WSN in Asynchronous Sleep Mode

ZHANG Zhi-hua[1,2], LUO Shou-shan[1,2], ZHU Hong-liang[1,2], XIN Yang[1,2,3]

(1. School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China;
2. National Engineering Laboratory for Disaster Backup and Recovery, Beijing University of Posts and Telecommunications, Beijing 100876, China;
3. Network Security Institute, Beijing Safe-Code Technology Co., Ltd, Beijing 100082, China)

Abstract: In order to detect the captured nodes as early as possible, just before they rejoin the networks, a node capture early detection method was proposed for wireless sensor networks (WSN) in asynchronous sleep mode. The proposed scheme was based on communication monitoring between neighbor nodes and a Hello message scheduling mechanism was employed to ensure the reception of Hello messages between asynchronous sleep nodes. Moreover, the local collaborative decision of common neighbors was explored to improve the detection accuracy rate. Simulations show that the proposed scheme outperforms other typical methods in terms of effectiveness including detection rate, false positive rate and false negative rate.

Key words: wireless sensor networks; physical capture; node compromise; asynchronous sleep; early detection

无线传感器网络因具备自组织、便捷性和低成本的特点,广泛应用于物联网感知层,适用于智能家居、健康监控、环境监测和战场数据收集等场景[1-3]. 为了使收集的数据更准确,网络一般采用节点冗余部署方式,同时节点采用休眠/唤醒机制运行[4],以降低能量消耗. 在应用中传感器节点一般部署在较为恶劣的开放式环境中,且限于成本,节点一般未配备物理防篡改装置[5]. 因此,环境和无线通信的开放性使节点处于安全威胁当中. 攻击者可以通过物理捕获传感器节点,获取节点中存储的所

收稿日期: 2017-11-04

基金项目: 国家重点研发计划项目(2017YFB0802300); 国家高技术研究发展计划(863 计划)项目(2015AA017201); 国家自然科学基金项目(U1536119); 广东省应用型科技研发专项资金项目(2015B010131007)

作者简介: 张志华(1984—), 男, 博士生, E-mail:zhangzhihua@bupt.edu.cn. 罗守山(1962—), 男, 教授, 博士生导师.

有关键机密数据，并可修改其程序代码[6]，控制传感器节点对网络发起内部攻击，从而控制或破坏网络.

1　相关工作

攻击者对无线传感器网络的攻击一般分为 3 个阶段[7-8]：①物理捕获一些节点，破解并获得其存储的所有关键机密数据；②将被捕获节点或其克隆节点重新加入原网络通信；③控制被捕获节点发起各种内部攻击. 现有研究多针对后 2 个阶段的攻击进行检测，包括节点重新部署位置检测[9]、节点克隆检测[5,10]、选择性转发攻击检测[11]、黑洞攻击检测[12]、女巫攻击检测[13]、DoS(Denial of Service)攻击检测和恶意数据注入检测[14]等. 为了减少这些通过被捕获节点对网络造成的危害，被捕获节点应该尽可能早地被检测出来. 理想情况下，节点刚被捕获就应被发现，并将其从网络通信中隔离[15]，指的就是对第 1 阶段节点物理捕获攻击的早期检测.

节点物理捕获攻击主要利用传感器的编程和测试接口，如 JTAG(joint test action group)接口(用于芯片测试和系统在线编程)对传感器进行妥协，达到控制的目的. 研究结果表明，节点捕获攻击需要将节点从正常网络通信中“移除”一段不可忽略的时间[16]，从几分钟到数小时不等. 现有对节点捕获攻击的检测方法正是利用这一特征，通过节点间广播信标(Beacon)消息[7]、问候(Hello)消息[8]或心跳(Heartbeat)消息[15]相互监控对方的存活性，以尽早发现被捕获节点.

基于配对的节点捕获检测(CAT, couple-based node compromise detection)方法[7]采用节点两两配对方式，通过发送接收 Beacon 消息相互监控对方的存活性以发现节点捕获，但是如果配对的节点同时被捕获，则无法检测到. 汇聚节点增强的第 1 阶段检测(SEFSD, sink enhanced first stage detection)方法[8]通过节点定期广播 Hello 消息以证明自己的存活，若监控节点未收到来自被监控节点连续的 Hello 消息超过一定数量，则认为其被捕获. 早期节点捕获检测(ENCD, early node compromise detection)方法[15]可对分簇网络中不同的簇，自适应采用不同的 Hello 消息广播频率以节约能量. 但以上方法存在 2 个不足之处：一是均未考虑节点休眠机制，若节点处于休眠状态，则会错过邻居节点广播的存活性声明消息，会造成很高的误报；二是监控节点仅通过自己是否收到被监控节点的声明消息来决策，未向其他邻居节点询问确认，也会造成较高的误报. 虽然 Ding 等[17]对节点捕获攻击的检测考虑了节点的休眠机制，但是其限定节点须处于同步休眠状态；Hsin 等[18]在节点监控决策过程中实现了本地一致性决策，但是其未考虑休眠机制. 笔者所提方法对同步或异步休眠模式下的节点捕获攻击进行检测，属于 3 个攻击阶段的第 1 阶段早期检测.

2　网络模型和假设

针对典型无线传感器网络中的节点物理捕获攻击进行检测. 传感器网络由一个基站(BS, base station)和大量传感器节点组成，假设传感器节点集合 $N=\{N_1, N_2, \cdots, N_n\}$，各节点随机均匀分布在目标区域中，其中 BS 负责收集传感器节点上报的数据，并向节点发送相关控制命令，它在网络中是可信和安全的. 每个节点在网络部署后均为静止的，且具备一个全网唯一的 ID 标识. 节点部署后发现、记录自己的一跳邻居，并在初始化路由形成过程中上报给 BS，各相邻节点互通彼此邻居列表. 节点周期性地将感知数据通过多跳路由发送至 BS. 各节点采用相同的休眠调度周期，但是无需同步休眠、同步唤醒，也无需时钟同步. 假设节点在一个休眠周期内，工作时长为 p_a，休眠时长为 p_s，且 $p_a > p_s$，p_a 值的选择要小于节点捕获攻击所需要的时长 T_{capture}，可根据不同的安全需求设定不同的值. 攻击者可以同时捕获一部分节点，假设有 n 个节点，攻击者可以同时捕获 $m(m<n)$ 个节点.

3　节点捕获早期检测方法

3.1　方法概述

采用邻居节点存活性主动监控方法，在节点休眠周期内的工作期间对其存活性进行监控. 每个节点主动监控其邻居节点，同时又被其邻居节点所监控. 监控节点自身由休眠状态唤醒时，为每个邻居节点设定分步计时器 T_1 和 T_2. 如果监控节点在 T_1 时间内未收到来自被监控节点的 Hello 消息或其他任何消息，则启动 T_2 计时器，同时向其他邻居询问对同一被监控节点的判断情况，根据邻居反馈情况做出协同决策，最终根据被监控节点的异常系数判断其是否从正常网络通信中消失，即是否被捕获.

一个节点 i 广播的 Hello 消息格式为〈ID_i, Neighbor_i, r_i〉，记作 Hello(i)，其中 ID_i 为节点 i 的

ID, Neighbor_i 为节点 i 的邻居表, r_i 为节点 i 距离它本次进入休眠状态的剩余时间.

3.2 节点声明消息统一调度广播

若节点处于休眠状态,则无法接收和发送包括 Hello 消息在内的所有消息,监控节点因此会造成对节点的误报. 为了解决节点周期性休眠导致的误报问题,采用节点声明消息统一调度广播机制,以保证其邻居节点在每个休眠周期内至少存在 1 次收到其 Hello 消息的机会. 文中所指的广播如未经特别说明均指的是向 1 跳邻居范围广播. 本方法设计的调度广播机制为在每个工作周期内主动广播 3 次 Hello 消息:第 1 次为节点由休眠状态唤醒为工作状态后;第 2 次为节点唤醒后经过一段长度为 p_s 的时间后;第 3 次为节点进入休眠状态之前,即唤醒后经过一段长度略小于 p_a 的时间后. 声明消息统一调度广播机制如图 1 所示.

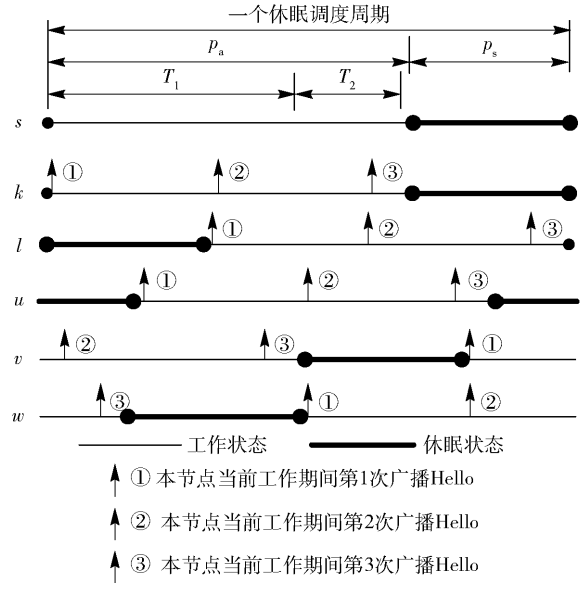

图 1 一个休眠周期内 Hello 消息调度广播计划

图 1 中,假设 s 节点为监控节点, k、l、u、v、w 节点为 s 的邻居节点,是相对于 s 而言的被监控节点. 各节点每个休眠周期时间长度相同,但是状态可能与监控节点同步(如 k 节点),也可能异步(如其他节点). s 节点由休眠状态唤醒时,为每个邻居节点设定分步计时器,各被监控节点按照声明消息统一调度机制主动广播 Hello 消息,由于休眠异步,在图 1 所示的当前状态,s 节点在当前工作周期内可以收到来自于 k 节点的 3 个 Hello 消息和来自于其他每个节点的 2 个 Hello 消息,但是 v 节点每个工作周期只能接收到来自于 u 节点的 1 个 Hello 消息.

3.3 详细检测过程

节点捕获早期检测在全网节点分布式运行,并且每个节点在其休眠周期内执行检测任务,根据监控和本地咨询的结果为被监控节点做出本地协同决策,当决策为异常时,通过多跳路由向 BS 发出报警信息,由 BS 定期通告全网,并采取必要措施.

节点在监控过程中为每个被监控节点 i 维护 3 个信息:分步计时器 $T_1(i)$ 和 $T_2(i)$、状态信息 $S(i)$ 以及异常指数 $A(i)$,同时有一个系统设定的异常告警阈值 H. 假设计时器信息初值设定为 $T_1(i)=T_1$、$T_2(i)=T_2$;状态信息指的是监控节点认为被监控节点当前所处的状态,状态集合为 $S=\{$被监控, 被审查, 正常, 怀疑, 异常$\}$,其中正常、怀疑和异常是节点检测的结果状态;异常指数为节点异常的可能性,取值范围为[1,10],取值越大说明其异常的可能性越大,其初值 $A(i)=5$. 监控节点在每个休眠周期进入休眠之前会确定每个被监控节点的结果状态.

监控节点对被监控节点的状态判断及协同决策过程如图 2 所示,图中采用 $\frac{C}{A}$ 形式表示状态转换的条件(C)和 s 节点采取的行动(A).

被监控状态:监控节点 s 唤醒时,将每个邻居节点 i 设置为被监控状态即 $S(i)=$被监控,同时为每个邻居节点启动计时器 $T_1(i)=T_1$,用于监控邻居节点 i 广播的 Hello(i)消息. 此时,可能有以下 4 种情况.

1) 若在 $T_1(i)$ 内,s 收到来自 i 的任何消息,包括 Hello(i)和其他任何数据消息,s 将 i 的状态转换为正常,同时关闭对 i 的计时器 $T_1(i)$,并重置异常指数 $A(i)$ 为初值.

2) 若在 $T_1(i)$ 内,s 未收到任何来自于 i 的消息,则将 i 的状态转换为被审查,并启动 $T_2(i)=T_2$,同时向邻居广播对 i 的咨询消息 consult(i),格式为〈CST, ID_s, ID_i, r_s〉,其中 CST 代表咨询消息,ID_s 和 ID_i 分别表示节点 s 和 i 的 ID,r_s 表示节点 s 距离它本次进入休眠状态的剩余时间.

3) 若在 $T_1(i)$ 内,s 收到来自其他邻居节点发来的针对 i 的咨询消息 consult(i),则 i 的状态仍为被监控,同时回应正在监控消息 monitoring(i),格式为〈MTR, ID_s, ID_i, r_s〉,其中 MTR 代表监控消息,即 s 正在对 i 进行监控.

4) 若在 $T_1(i)$ 内,s 收到来自任一邻居节点发来的针对 i 的异常广播消息 abnormal(i),格式为

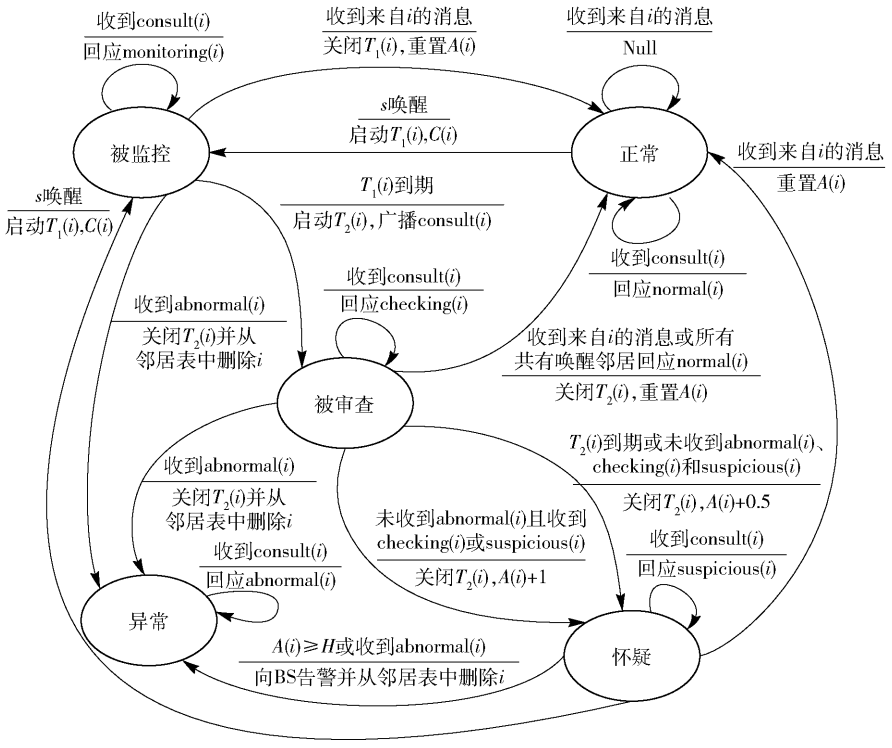

图 2　监控节点 s 认为被监控节点 i 所处状态及转换

〈ABN, ID_s, ID_i, r_s〉,其中 ABN 代表异常,则将 i 的状态转换为异常,关闭 $T_1(i)$,并将 i 从邻居列表删除.

正常状态:当节点 s 认为节点 i 处于正常状态时,s 节点在本次调度周期内不再对 i 的消息计时,收到来自 i 的消息时,其状态仍保持正常. 若 s 收到来自其他邻居节点关于 i 的咨询消息 consult(i),则回应一个正常消息 normal(i),格式为〈NML, ID_s, ID_i, r_s〉. 当 s 节点本身由休眠状态唤醒时,它将 i 的状态设置为被监控,同时启动计时器 $T_1(i) = T_1$.

被审查状态:当监控节点 s 将邻居节点 i 设置为被审查状态时,s 向邻居广播对 i 的咨询消息 consult(i),并在计时器 $T_2(i) = T_2$ 时间内等待处于唤醒状态的共同邻居对这个咨询消息的回应,根据回应消息对节点 i 的状态进行协同决策. 若在此状态,s 收到其他邻居节点发来的咨询消息 consult(i),则回应 checking(i),格式为〈CHK, ID_s, ID_i, r_s〉,其中 CHK 代表正在审查消息. 当节点收到 consult(i)时,需要根据自己对节点 i 的判断状态进行回应,回应消息包括 monitoring(i)、normal(i)、abnormal(i)、checking(i) 和 suspicious(i). 当节点 i 自己收到 consult(i)时,须回应 Hello(i)消息以声明自己的存活. s 根据回应消息判断 i 的状态,包括以下 5 种情况.

1) 若 s 收到来自 i 的任何消息,包括 Hello(i)和其他任何消息,或收到所有唤醒的共同邻居回应 normal(i),则 s 将 i 的状态转换为正常,同时关闭对 i 的计时器 $T_2(i)$,并重置异常指数 $A(i)$ 为初值.

2) 若 s 未收到来自 i 的任何消息,但收到来自任一邻居发来的 abnormal(i),则将 i 的状态转换为异常,同时关闭 $T_2(i)$ 并将 i 从邻居列表中删除.

3) 若 s 未收到 abnormal(i),但收到 checking(i)或 suspicious(i),则将 i 的状态转换为怀疑,关闭计时器 $T_2(i)$,并增加 i 的异常系数 $A(i) = A(i) + 1$. 这说明邻居节点中有其他节点也对 i 的存活性有异议,因此节点 i 的异常指数显著上升.

4) 若 s 未收到 abnormal(i)、checking(i) 和 suspicious(i),但收到 monitoring(i),则将 i 的状态转换为怀疑,关闭计时器 $T_2(i)$,并增加 i 的异常系数 $A(i) = A(i) + 0.5$. 这说明有邻居节点正处于对 i 的存活性监控状态,还未得到结论,节点 s 需根据自己的信息判断节点 i 的状态,因 s 未收到来自节点 i 的信息,故 i 的异常指数小幅上升.

5) 若 $T_2(i)$ 到期,s 仍未收到任何消息,则将 i 的状态转换为怀疑,并增加 i 的异常系数 $A(i) = A(i) + 0.5$. 这表明其邻居节点可能在休眠,s 节点

需自己判断 i 的状态,因此 i 的异常指数小幅上升.

怀疑状态:当节点 s 认为节点 i 进入怀疑状态时,它首先检查 i 的异常系数是否达到了告警阈值 H,若达到阈值,则将 i 的状态转换为异常,并向 BS 发出告警信息,同时将 i 从自己的邻居列表中删除. 若在此状态,s 节点收到了来自于 i 的任何消息,则将 i 的状态转换为正常,并重置 i 的异常系数为初值. 若 s 节点收到其他节点关于 i 节点的咨询消息 consult(i),则回应怀疑消息 suspicious(i),格式为〈SUS, ID_s, ID_i, r_s〉,其中 SUS 代表怀疑消息. 当节点 s 本身由休眠状态唤醒时,它将 i 的状态转换为被监控,同时启动新一轮的计时器.

异常状态:当 s 认为节点 i 处于异常状态,即认为 i 节点被捕获时,它将通过多跳路由向 BS 告警,同时将 i 从自己的邻居列表中删除. BS 会定期将被捕获节点在全网范围通告. 若在 BS 通告之前,s 节点收到邻居节点发来的针对 i 的咨询消息 consult(i),则回应 abnormal(i).

3.4 方法分析

为保证检测方法有效,时间参数设定需满足:

$$T_{\text{capture}} > p_a > T_1 + T_2, T_1 > p_a - p_s \text{且} T_1 > p_s > T_2 \tag{1}$$

异步休眠模式下对节点捕获攻击进行检测,其关键是保证在节点唤醒状态即工作周期内有收到被监控节点 Hello 声明消息的机会,防止因休眠状态造成的误报. 因此,监控节点设定的分步计时器 $T_1 + T_2$ 应在其工作周期之内,以实现对被监控节点做出判断,即 $p_a > T_1 + T_2$. 为了保证被监控节点经过 p_s 长度的休眠后唤醒时,监控节点有机会收到其 Hello 消息,节点工作时长应大于休眠时长,即 $p_a > p_s$,同时为减少不必要的咨询消息而造成的能量消耗,T_1 计时器的长度应大于 3 次统一调度 Hello 消息之间的间隔 p_s 和 $p_a - p_s$,即 $T_1 > p_s$ 和 $T_1 > p_a - p_s$,又因为 $p_a > T_1 + T_2$,所以 $T_1 > T_1 + T_2 - p_s$,即得到 $p_s > T_2$. 如图 1 所示,若节点 k 监控节点 l,则节点 k 设定的 T_1 应大于 p_s;若节点 l 监控节点 k,则节点 l 设定的 T_1 应大于 $p_a - p_s$. 同时,应保证节点休眠调度周期的长度小于节点捕获所需要的时间 T_{capture},以使节点在重新加入网络通信之前被检测出来.

4 实验仿真

下面对所提节点捕获检测方法的有效性和通信开销情况进行评价. 实验采用 OMNET + + 进行仿真,并在相同条件下与典型节点捕获攻击检测算法 CAT[7] 和 SEFSD[8] 进行比较.

实验在 200 m × 200 m 的场地中随机均匀部署 200 个节点和 1 个 BS,每个节点的通信范围相同均为 40 m,实验设定捕获节点的最高比例为 25%. 为了不失一般性,实验设定选择 5 min 的 1/3 约 95 s 为节点捕获所需的时间,每个节点休眠调度周期均为节点捕获所需时间的 1/4,即 23 s,其中工作周期 p_a 为 13 s,休眠周期 p_s 为 10 s,监控节点为每个邻居节点设定的分步计时器 T_1 为 11 s,T_2 为 1.5 s;在监控节点判断被监控节点有连续两次明显怀疑状态的情况下,认定其为异常,因此节点的异常系数阈值 H 为 7.

采用 3 种方法的被捕获节点检测率如图 3 所示,检测率均随被捕获节点数量的增加而下降. 其中,CAT 方法检测率下降最为明显,从 100% 下降到 75.1%;SEFSD 方法次之,从 100% 下降到 85.6%;所提方法检测率下降最为平缓,从 100% 下降到 95.2%. 检测率下降的原因主要是随着被捕获节点数量增加和误报数量增加,监控节点的数量有所下降,有些节点无法被监控,导致检测率下降. CAT 方法检测率下降明显的原因,一是由于 CAT 采用节点配对互相监控,存在配对节点同时被捕获的情况,随着被捕获节点数量增加,配对节点同时被捕获的概率也随之增加;二是对异步休眠节点的误报导致监控节点数量的快速下降,故其检测率的下降也相对更加明显. 对 SEFSD 方法来说,由于未考虑节点休眠的影响,其检测过程受到节点异步休眠的干扰,造成一部分正常节点被误认为是被捕获节点而从网络中移除,导致监控节点数量减少,出现有些节点没有其他节点监控的情况,因此其检测率会随之下降. 所提方法在检测过程中避开了节点休眠周期的影响,能够保证在节点的工作周期收到邻居节点的声

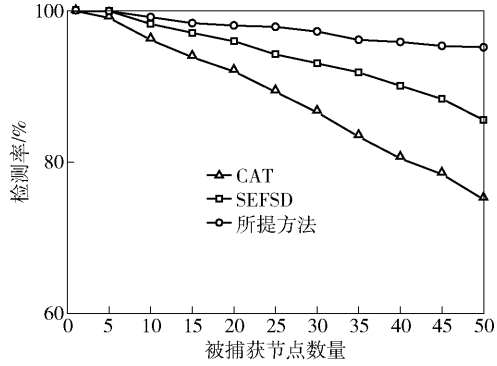

图 3 3 种方法节点捕获检测率

明信息，因此其检测率下降最为平缓.

采用 3 种方法进行捕获节点检测的误报率均呈现上升趋势，如图 4 所示. 其中，SEFSD 方法的误报率上升最为明显，从 19.1% 上升到 31.7%；CAT 方法次之，由 12.7% 增加到 25.3%；所提方法误报率相对是最低的，从 1.1% 到 8.2%. 造成误报的原因一方面是节点异步休眠导致的，被监控节点休眠状态下无法及时发送声明消息，因此监控节点无法收到连续的 Hello 信息，造成对处于休眠的正常节点的误报；另一方面是 CAT 和 SEFSD 方法在检测过程中，只是根据自己收到数据的情况进行判断，容易因为丢包等情况造成判断失误，所以也会增加一定的误报率. 由于 CAT 方法 1 个节点仅监控配对的 1 个节点，而 SEFSD 方法 1 个监控节点需监控多个邻居节点，在异步休眠的情况下，其误报会更多. 所提方法一方面排除节点异步休眠的影响，另一方面采用节点协同决策的方案，增加询问环节，既考虑节点本身收到数据的情况，又对怀疑情况向共同邻居节点进行询问，提升了决策的准确度，因此误报相对其他方法较低.

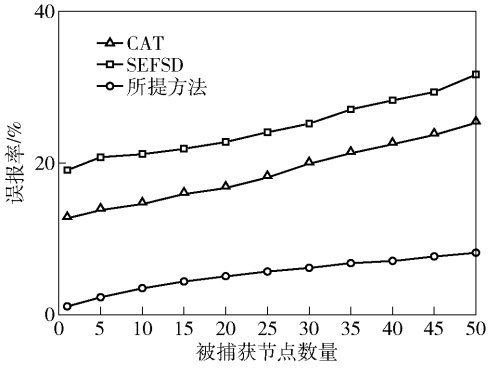

图 4　3 种方法节点捕获检测误报率

采用 3 种方法进行捕获节点检测的漏报率情况如图 5 所示. 漏报率最高的是 CAT 方法，由 0 增加到 24.5%；其次是 SEFSD 方法，由 0 增加到 13.9%；所提方法漏报率呈较低水平，由 0 增加到 5.1%. 漏报率主要是由被捕获节点数量增加，加之误报数量增多造成节点的监控数量下降而造成的. CAT 方法的配对节点有同时捕获和误报的风险，因此其漏报率明显高于其他方法. 所提方法由于误报持续在较低的水平，监控节点不会显著减少，同时将网络边缘等某些孤立特殊节点考虑进来，由 BS 根据其邻居捕获情况进行判断，因此漏报数量较其他方法维持在相对较低的水平.

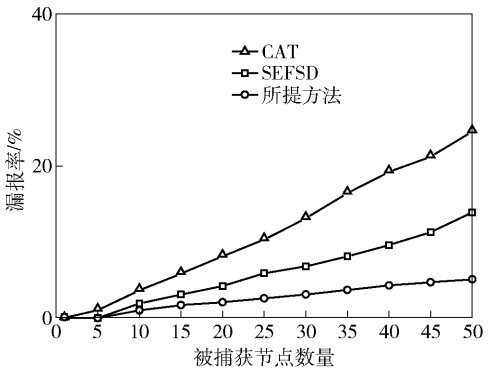

图 5　3 种方法节点捕获检测漏报率

在检测捕获节点过程中，网络需承担一定的通信负载，考虑每个休眠周期检测过程网络所有节点需发送的检测消息数量，包括广播 Hello 消息、咨询消息、回复消息以及告警消息. 在相同条件下，网络通信负载情况如图 6 所示，其中通信负载量最大的是 SEFSD 方法，其次是所提方法，最小的是 CAT 方法.

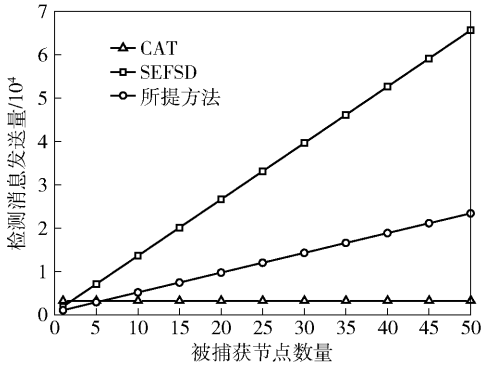

图 6　3 种方法网络通信负载

通信负载存在较大差异主要是各方法的检测机制不同造成的. SEFSD 方法在每个周期中定期广播 3 个 Hello 消息，若监控节点连续 3 次未收到邻居节点的 Hello 消息，则广播 2 个探测消息，所有收到探测消息的节点需回复存活性证明消息，因此每个周期中网络的检测信息发送量为 $3n + 2md + 2md^2 + m$，其中 n 为节点总数，m 为被捕获节点数量，d 为邻居节点平均数量(文中仿真条件下 $d = 25$). 所提方法在每个周期内固定广播 3 个 Hello 消息，若在 T_1 计时器内监控节点未收到邻居节点的 Hello 消息，则向邻居节点广播咨询消息，但是只有监控节点与被监控节点的共同邻居节点才回复这个咨询消息，因此每个周期中网络的检测信息发送量为 $3n + md + 0.688\,5md^2 + m$，其中 $0.688\,5d$ 为 2 个节点间共同邻居的数量[18]. 而对于 CAT 方法，由于其仅仅

是2个配对节点之间进行监控和通信,检测过程中不涉及其他邻居节点,因此其检测信息发送量只包括2个节点配对和监控所发送的信息,其总量为$6.5n+m$,故而其通信负载在3种方法中是最小的,但是这种方法牺牲了检测性能,有很高的漏报率.因此,所提方法在提高检测效果的情况下,通信负载比SEFSD方法显著降低.

5 结束语

针对现有无线传感器网络节点捕获攻击检测方法未考虑节点异步休眠的情况,提出异步休眠模式下节点捕获攻击早期检测方法,以实现网络攻击的早期发现. 该方法通过节点声明消息统一调度广播机制确保监控节点对邻居节点声明消息的接收,并采用本地协同决策机制提高检测的正确率. 仿真结果表明,所提方法有效避免了节点异步休眠对捕获攻击检测的影响,提高了节点捕获攻击的检测率,并显著降低了漏报率和误报率.

参考文献:

[1] Zhang Yongmin, He Shibo, Chen Jiming. Data gathering optimization by dynamic sensing and routing in rechargeable sensor networks[J]. Sensor, Mesh & Ad Hoc Communications & Networks, 2013, 24(3): 273-281.

[2] Hu Yanling, Liu Anfeng. An efficient heuristic subtraction deployment strategy to guarantee quality of event detection for WSNs[J]. Computer Journal, 2015, 58(8): 1747-1762.

[3] Jiang Liangshan, Liu Anfeng, Hu Yanling, et al. Lifetime maximization through dynamic ring-based routing scheme for correlated data collecting in WSNs[J]. Computers & Electrical Engineering, 2015, 41(1): 191-215.

[4] Abdelsalam H S, Olariu S. Toward adaptive sleep schedules for balancing energy consumption in wireless sensor networks[J]. IEEE Transactions on Computers, 2012, 61(10): 1443-1458.

[5] Shashidhar N, Kari C, Verma R. The efficacy of epidemic algorithms on detecting node replicas in wireless sensor networks[J]. Journal of Sensor & Actuator Networks, 2015, 4(4): 378-409.

[6] Lin Chi, Wu Guowei, Yu Chang Wu, et al. Maximizing destructiveness of node capture attack in wireless sensor networks[J]. Journal of Supercomputing, 2015, 71(8): 3181-3212.

[7] Lin Xiaodong. CAT: building couples to early detect node compromise attack in wireless sensor networks[C]// Global Telecommunications Conference. Honolulu: IEEE Press, 2009: 1-6.

[8] Ding Wei, Yu Yingbing, Yenduri S. Distributed first stage detection for node capture[C]// Global Telecommunications Conference. Miami: IEEE Press, 2010: 1566-1570.

[9] Song Hui, Xie Liang, Zhu Sencun, et al. Sensor node compromise detection: the location perspective[C]// International Conference on Wireless Communications and Mobile Computing. Honolulu: ACM Press, 2007: 242-247.

[10] Zeng Yingpei, Cao Jiannong, Zhang Shigeng, et al. Random-walk based approach to detect clone attacks in wireless sensor networks[J]. IEEE Journal on Selected Areas in Communications, 2010, 28(5): 677-691.

[11] Ren Ju, Zhang Yaoxue, Zhang Kuan, et al. Adaptive and channel-aware detection of selective forwarding attacks in wireless sensor networks[J]. IEEE Transactions on Wireless Communications, 2016, 15(5): 3718-3731.

[12] Liu Yuxin, Dong Mianxiong, Ota K, et al. ActiveTrust: secure and trustable routing in wireless sensor networks[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(9): 2013-2027.

[13] Abbas S, Merabti M, Llewellyn-Jones D, et al. Lightweight sybil attack detection in MANETs[J]. IEEE Systems Journal, 2013, 7(2): 236-248.

[14] Nam S M, Cho T H. Context-aware architecture for probabilistic voting-based filtering scheme in sensor networks[J]. IEEE Transactions on Mobile Computing, 2017, 16(10): 2751-2763.

[15] Al-Riyami A, Zhang Ning, Keane J. An adaptive early node compromise detection scheme for hierarchical WSNs[J]. IEEE Access, 2016, 4(1): 4183-4206.

[16] Becher A, Benenson Z, Dornseif M. Tampering with motes: real-world physical attacks on wireless sensor networks[C]// Third International Conference on Security in Pervasive Computing. York: ACM, 2006: 104-118.

[17] Ding Wei, Yu Yingbing, Yenduri S. Energy saving by centralized sleep in early detection of captured nodes [C]// IEEE International Workshop on Robotic and Sensors Environments. Phoenix: IEEE Press, 2010: 1-6.

[18] Hsin C, Liu Mingyan. Self-monitoring of wireless sensor networks[J]. Computer Communications, 2006, 29(4): 462-476.