

选择传递攻击中的异常丢包检测

俞 波 杨 珉 王 治 高传善

(复旦大学计算机科学与工程系 上海 200433)

摘 要 该文提出了一种基于检查点的多跳确认方案来检测选择传递攻击所导致的异常丢包. 在这个方案中, 能够随机地选取传递路径中的部分节点为检查点, 负责包的确认. 这种随机检查点选择技术能够避免部分节点成为敌方俘获的目标. 从而在保证检测能力的同时有效地提高了系统的健壮性. 文章对检测率进行了理论分析, 并进一步通过模拟实验对检测率进行了验证.

关键词 无线自组网络; 无线传感器网络; 入侵检测; 选择传递攻击; 安全路由

中图法分类号 TP393

Identify Abnormal Packet Loss in Selective Forwarding Attacks

YU Bo YANG Min WANG Zhi GAO Chuan-Shan

(Department of Computer Science and Engineering, Fudan University, Shanghai 200433)

Abstract In this paper, the authors propose a lightweight security scheme, CHEMAS (CHECKpoint-based Multi-hop Acknowledgement Scheme), for detecting selective forwarding attacks. This scheme can randomly select part of intermediate nodes along a forwarding path as checkpoint nodes which are responsible for generating acknowledgements for each packet received. The strategy of random checkpoint selection significantly increases the resilience against attacks because it prevents a proportion of the sensor nodes from becoming the targets of attempts to compromise them. The authors examine the detection accuracy of this scheme using both theoretical analysis and simulations.

Keywords Ad Hoc networks; wireless sensor networks; intrusion detection; selective forwarding attacks; secure routing

1 引 言

无线传感器网络在环境监视相关的领域都有着广阔的发展前景, 例如, 军事监视、森林火灾检测等. 在这些应用中, 数量众多的传感器节点被部署到一片辽阔的区域, 然后对部署区域中发生的特定事件进行检测 (例如, 敌方坦克的移动、森林火灾的爆发). 一旦相应事件被检测到, 节点会生成报告, 并以多跳 (multi-hop) 方式通过无线链路将报告递交给

基站 (base station, 又称 sink 节点). 在这类应用中, 信息迅速、准确、无误的传递对整个任务的成败起着关键作用, 我们称这类传感器网络应用为具有重要使命的应用. 这也是本文提出方案所针对的应用环境.

由于传感器节点本身的脆弱性, 这类传感器网络很容易收到一系列不同类型的恶意攻击, 其中之一就是选择传递攻击 (selective forwarding attacks). Karlof 最先提出了这种传感器网络中的攻击方式^[1]. 如图 1 所示, 在这种攻击中, 恶意节点在大多

数时间表现像正常节点一样,然而它们会选择性地丢弃一些敏感的包,例如,一个报告敌方坦克移动的包.通常,如果恶意节点正好处于数据传递的路径上时,选择传递攻击会变得更为有效.特别当与其它攻击相结合后,例如,虫洞攻击(worm hole)、污水池攻击(sinkhole),选择传递攻击会变得更有破坏力,能够破坏一系列路由协议的正常运作,包括 TinyOS beaconing, Directed Diffusion^[2], GPSR^[3], GEAR 以及基于簇的路由协议.

目前,国内外针对防范传感器网络选择传递攻击的研究很少.有一种可以缓解选择传递攻击影响的方法是多路径传递(multipath forwarding). Karlof 在文献[1]中提到了这种方法,但他并没有提供详细的方案.多路径传递利用冗余性提高成功递交率,但也具有不少缺陷:首先,通信开销与路径数量成正比.随着路径数量的增加,总的通信开销迅速上升.其次,多条路径终将在靠近基站(base station)的位置合并为一条路径,因此在基站附近仍然可以实施有效的选择传递攻击.另外,多路径传递能达到的安全性也十分有限.只要每个路径上有一个节点被俘获,多路径传递的目的也彻底失败.

本文中,我们扩展了前期的工作^[4]并提出了一种基于检查点的多跳确认方案(Checkpoint-based Multi-hop Acknowledgement Scheme),这是一个利用随机选择的检测点进行多跳确认的入侵检测方案.本文中,一条传递路径上的部分节点会被随机地选取为检查点,检测点会为它收到的每个事件包(event packet)生成一个确认包(ACK packet),并将确认包向上游传递.任何传递路径上的中间节点,如果没有收到足够的确认包,它会生成异常丢包的警告信息,并经过多跳递交给源节点.本文中,源节点具有搜集异常丢包信息的能力,而很多传统传感器网络入侵检测算法是在基站或者其它中心机构实现的.源节点能够搜集异常信息带来的好处是即使基站或者其它中心机构在攻击中失效时,源节点仍然能够进行检测并做出适当决策.模拟实验表明即使在信道误码率在 15% 时(通常被认为是十分恶劣的信道条件),我们的异常丢包检测率仍然能够达到 95%.而在通信开销方面,不论攻击是否存在,通信开销也仅为单路径传递的 1.5 倍左右.

2 预备工作

在这一节中,我们首先定义了本文方案所适

用的假设,接着介绍了本文提出的位置绑定 ID 密钥技术,然后简单描述了本文方案所直接采用的 μ TESLA 协议^[5],最后提出了我们对总体防御步骤的建议.

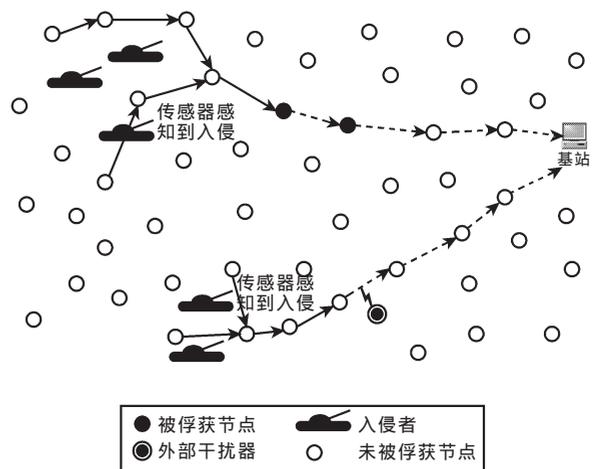


图 1 受到选择传递攻击的无线传感器网络

2.1 假设

本文提出的检测方案针对具有重要使命的传感器网络应用(例如战场监视),其应用环境与民用传感器网络(例如气温监测)有很大差异,以下若干前提条件将适用于本文提出的检测方案:(1)我们假定在部署阶段每个传感器节点能获取自己的地理位置,并与基站保持松散时间同步.安全定位和时间同步也是军事应用最基本的需求(文献[1,6,8]中讨论了相关问题),所以本方案并不因该假设增加额外开支.(2)我们假定在部署阶段,传感器网络处于完全安全状态,敌方无法俘获我方节点.已有的传感器网络安全协议^[6]同样持有该假设,并证明在实际操作中如果采用了适当的保护措施和部署计划,这一点是容易实现的.(3)我们假定在传感器节点上已实现路由协议和传输层协议(例如,Directed Diffusion^[2],GPSR^[3],PSFQ^[7]),我们提出的安全协议可以运作在这些协议之上,但并不依赖于具体协议.

有多种针对传感器网络路由层的攻击,例如,编造和篡改数据(spoofed and altered packets)、女巫攻击(sybil Attacks)、虫洞攻击(wormhole attacks)、污水池攻击(sinkhole attacks)等,但本文不考虑针对这些攻击的防范措施.有兴趣的读者可以参考文献[1,6,8].

2.2 位置绑定 ID 密钥技术

在这一小节中,我们介绍本文提出的位置绑定 ID 密钥技术(location-binding ID-based key man-

agement). 该技术能够帮助我们轻松地推导出节点的位置信息, 并且能够帮助网络中的任意两节点建立会话密钥, 从而实现数据验证、保密通信等目的. 位置绑定 ID 密钥是构成本文提出的选择传递攻击检测方案的重要技术之一.

传感器节点部署到监测环境中后通常位置是固定的, 因此我们没有必要为每个节点分配一个传统的节点序号. 许多基于位置信息的路由协议(例如, GPSR^[3])实际上也直接将节点的位置信息作为节点 ID. 因此, 我们完全可以将节点的位置坐标作为节点 ID, 这样只要知道节点的 ID 就可以在一定精度上知道节点位置. 一个重要的问题是如何防止恶意节点伪造他们的 ID 和位置信息? 我们通过双变对称多项式来实现基于位置绑定 ID 的密钥管理技术.

我们的位置绑定 ID 密钥管理通过以下两步实现: 第 1 步, 在部署前, 密钥服务器在有限域 F_q 生成

一个对称双变多项式 $f(u, v) = \sum_{i,j=0}^k a_{i,j} u^i v^j$, 其中 q

是一个足够大的质数能够容纳加密密钥. 我们称一个多项式是对称的, 如果 $f(u, v) = f(v, u)$. 密钥服务器为每个节点加载这个对称多项式. 第 2 步, 在部署阶段, 当每个节点通过安全定位算法获取了自己的位置坐标后, 每个节点将自己的位置信息作为节点的 ID: $ID = x \parallel y$, 其中 \parallel 表示连接操作. 接着, 每个节点生成一个新的多项式 $g(v) = f(ID, v)$, 并在内存中擦除多项式 $f(u, v)$. 注意, 每个节点中的 $g(v)$ 多项式是不同的, 是由节点的位置信息决定的.

下面我们举例说明位置绑定 ID 密钥管理技术是如何工作的. 假定在获取位置信息后, 节点 a 拥有 $g_a(v) = f(ID_a, v)$, 节点 b 拥有 $g_b(v) = f(ID_b, v)$, 节点 a 和 b 在互相的通信距离内, 节点 a 要发一个包给节点 b . 首先, 节点 a 计算 $k_1 = g_a(b)$ 作为会话密钥, 并用 k_1 加密数据包及生成 MAC 验证码(Message Authentication Code). 然后, 数据包和验证码将一同发送到节点 b . 此时, 节点 b 计算 $k_2 = g_b(a)$ 作为解密密钥, 这里 $k_1 = k_2 = g_a(b) = g_b(a)$ 是由对称多项式的属性决定的. 这样节点 b 就可以对数据包进行解密以及通过 MAC 码验证该包是否真的来自节点 a , 同时节点 b 也可以获取节点 a 的位置信息, 因为 a 的 ID 即 a 的位置坐标.

本文提出的位置绑定 ID 密钥管理技术简单有效, 具有以下几点特色: (1) 位置与 ID 绑定, 利用对称多项式实现数据加密与验证, 同时又可轻松获得

节点位置信息. (2) 能防止节点复制攻击或者节点被恶意移动. 例如, 敌方俘获一个节点后, 可以复制该节点, 但无法将克隆节点部署到网络其它位置, 因为, 该节点内存中的 $f(u, v)$ 已被擦除, 而内存中的 $g(v)$ 是与该节点的位置绑定的, 即使克隆节点部署到网络其它位置, 那里未被俘获的节点也会拒绝它通信(因为它的原先位置在这些节点的信号距离之外). (3) 达到一定的安全强度. 在应用到密钥管理中, 双变对称多项式已被证明只要不到 k 个节点被俘获, 就可以达到绝对的安全^[9]. 也就是说, 敌方无法通过被俘获节点内存中的函数 $g(v)$ 来推导出 $f(u, v)$ 从而达到编造任意节点 ID 的目的.

2.3 μ TESLA 安全多播协议的应用

本文直接采用 μ TESLA 协议^[5] 帮助在一条传递路径上的多节点间实现一对多的身份认证. 在这一小节中, 我们首先对 μ TESLA 协议做一简单介绍, 然后介绍如何将其应用到本文检测方案中.

μ TESLA 协议是在松散时间同步的基础上通过延迟公开密钥的方式实现认证广播的. 在该协议中, 发送者首先使用单向 Hash 函数 F 生成密钥链 $\langle k_0, k_1, \dots, k_n \rangle$ 并存储于内存中, 其中 $k_i = F(k_{i+1})$. 同时, 时间被分割为许多时间间隔(time intervals), 并且每一密钥与一段时间间隔相对应. 在 t_0 时刻, 发送者将 k_0 通过保密的方式(例如使用本文 2.2 节提出的位置绑定密钥技术)传递给所有接收者. 在 t_i 时刻, 发送者生成一个数据包, 并使用 k_i 对该数据包生成 MAC 码, 之后将该包发送给所有接收者. 在一定延时时, 即 $t_i + \delta$ 时刻, 发送者将密钥 k_i 公开给所有接收者. 因为, 接收者和发送者是时间同步的, 接收者即可通过单向 Hash 函数验证在 t_i 时刻接收的包是否来自真实的发送者. 利用 μ TESLA 可以方便地实现一对多传输时的数据认证, 节省计算与通信开销.

本文利用 μ TESLA 协议实现 ACK 包多播传输时的认证(我们会在第 3 节中具体介绍 ACK 包). 应用该协议时, 在部署阶段, 每个节点生成自己的单向 Hash 密钥链, 并将 k_0 发送给所有上游节点(本文中, 我们称在一条从源节点到基站的路径上, 靠近源节点的方向为上游, 靠近基站的方向为下游). 之后, 发送节点只要在相应时刻使用相应密钥链上的密钥对数据包生成 MAC, 即可实现一对多的数据认证. 例如图 3 中, 节点 u_4 生成了一个 ACK 包, 用密钥链上的相应时刻的密钥生成 MAC, 接着经过多跳发送给节点 u_3, u_2, u_1 . 这样 u_3, u_2, u_1 就可以通过

μ TESLA 延时泄密的机制验证 ACK 包是否被篡改, 或者是否来自真实的 u_4 节点。

μ TESLA 协议的运用会在一定程度上增加系统运作的延时, 但延时的程度可以根据具体应用的需求对相关参数进行一定调整, 从而实现性能均衡。如何将 μ TESLA 协议与本文方案进一步密切结合是我们未来工作的重要部分之一。

2.4 总体防御步骤

我们建议针对选择传递攻击的总体防御步骤可以包括以下三步: 预备阶段、入侵检测阶段和决策与反击阶段。

预备阶段。 即属于网络的部署阶段, 网络完成各项初始化工作, 包括常规方面的邻居发现、路由建立, 也包括安全性方面的安全定位、安全时间同步、相邻节点会话密钥建立(例如利用 2.2 节提出的位置绑定 ID 密钥技术)、 μ TESLA 协议密钥链的建立与 k_o 的交换等操作。

入侵检测阶段。 即网络行使其使命的阶段(例如监控战场)。入侵检测机制与网络常规路由传播协议协同运行, 在数据采集传播的同时完成对网络异常丢包的检测。异常丢包信息将被发送到源节点或者基站。

决策与反击阶段。 当基站或者源节点收集了足够多的网络异常丢包信息后, 可以通过运行更为复杂的 IDS(Intrusion Detection System) 算法来做出决策并进行反击, 例如, 可以通知路由协议调整路由、减少通过可疑节点的流量; 甚至可以派出士兵人工检查并移除恶意节点。

本文提出的基于检查点的多跳确认方案主要针对入侵检测阶段。

3 基于检查点的多跳确认方案

在这一节中, 我们详细介绍本文提出的基于检查点的多跳确认方案(checkpoint-based multi-hop acknowledgement scheme)。通过该方案, 我们可以检测出传递过程中不正常的丢包, 并发现导致丢包的可疑节点。检测丢包的一种简单方法是确认(acknowledgement), 无线传感器网络中的传输层协议例如 PSFQ^[7] 就使用基于跳(hop-by-hop)的确认来保证包的可靠传递。但是, 这类协议并不是为抵抗恶意攻击而设计的。受到传统基于跳的确认机制的启发, 我们可以使用多跳的确认技术来验证中间节点是否忠实地传递每个经过的包。

3.1 包的定义

在这一小节中, 我们仅简单给出应用在本文方案中的三个主要数据包的定义。这三个包的格式定义及建议的字段长度如图 2 所示。在 μ TESLA 协议中延时公开密钥需要的数据包, 本文限于篇幅不做介绍。

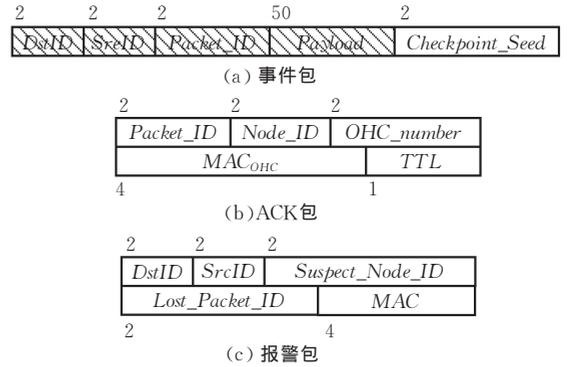


图 2 包格式定义

事件包(event packet)是当监测环境中相应事件发生时(例如敌方坦克移动事件)由源节点生成的包。源节点当然也可以根据基站的查询请求生成事件包。事件包会由源节点经过多跳传递到基站。如图 2(a)所示, 阴影部分的字段为常规路由所需要的字段, 而 *Checkpoint_Seed* 字段是由源节点设置的, 决定了路径上的哪些节点成为检查点。

ACK包(ACK packet)是由检查点(节点)生成的。检查点每收到一个事件包, 就生成一个 ACK 包, 并向上游传递。其中, *Packet_ID* 字段表示该 ACK 包需要确认的那个事件包 ID, *Node_ID* 是生成确认包的检查点 ID, *OHC_number* 和 *MAC_{OHC}* 是 μ TESLA 协议使用的两个包, *OHC_number* 是当前检查点的密钥链中选用的那个密钥, *MAC_{OHC}* 是使用该密钥生成的 MAC 码, *TTL* 指示了该包在传递过程中经过的检查点的数量。 *MAC_{OHC}* 字段的内容为

$$MAC_{OHC}\{Packet_ID, Node_ID, Payload\},$$

其中, *Packet_ID* 和 *Payload* 分别是事件包中的 *Packet_ID* 和 *Payload* 字段。 *MAC_{OHC}* 码的作用是为整个 ACK 包生成签名摘要, 防止恶意节点编造、篡改 ACK 包。

报警包(alert packet)是中间节点检测到异常网络丢包时生成的包, 将经过多跳传递到源节点。换句话说, 报警包也是最后一个“见到”先前事件包的中间节点产生的。当源节点接收到该包时, 源节点知道先前的事件包至少到达了产生该报警包的

中间节点. 在报警包字段中, $SrcID$ 表示该包的生成节点, $DstID$ 为该包的目的地节点(源节点 ID), $Lost_Packet_ID$ 为异常丢包的 ID, MAC 为通过 2.2 节位置绑定 ID 密钥技术生成的 MAC 码. MAC 字段的内容为

$$MAC\{DstID, SrcID, Suspect_Node_ID, \\ Lost_Packet_ID, Packet_ID, Payload\},$$

其中, $Packet_ID$ 和 $Payload$ 分别是事件包中的 $Packet_ID$ 和 $Payload$ 字段. 这里的 MAC 码同样起着防止恶意节点编造、篡改报警包的作用.

3.2 检测方案

在这小节中, 我们概述本文提出检测方案的基本原理.

本文检测方案基本思想如下: 在一条从源节点到基站的传递路径上, 部分节点被选择成为检查点(checkpoint nodes). 我们称路径上两个连续的检查点之间的中间节点构成了一个中间节点段. 这样, 一条传递路径即由若干检查点和中间节点段组成. 当源节点检测到一个特定的事件时(例如坦克移动噪音), 源节点生成一个事件包(event packet). 该事件包沿着路径, 经过多跳转发向基站传递; 转发事件包的中间节点会同时将事件包暂时保存在 Cache 中. 如果一个检查点从上游邻居节点收到了一个事件包, 在继续转发前该检查点会为该事件包生成一个 ACK 包(ACK packet), 并选择 μ TESLA 密钥链上当前的密钥对整个 ACK 包内容签名并生成一个

MAC 码. 之后, 该 ACK 包将沿着与先前事件包相同的路径但相反的方向向上游传递. 该 ACK 包在穿过了若干中间节点段后, 会被某个检查点或源节点抛弃. 因为 ACK 包上的 MAC 码可认证其来源, 这样, 所有 ACK 包所穿过的中间节点会相信先前的事件包已安全地到达了下游的检查点. 如果这些中间节点没有收到 ACK 包(或者没有收到足够数量的 ACK 包), 则很有可能先前的事件包已被某个被俘获节点恶意丢弃, 这样中间节点会生成报警包(alert packet)并传递给源节点. 报警包中, 该中间节点仅可指定其下游的直接邻居节点为可疑节点, 并使用 MAC 码对报警包内容签名. MAC 码的使用保证了报警包不可编造、不可篡改.

我们以图 3 为例进一步说明该检测方案. 在图 3 中, 我们假定节点 u_4, u_6, u_9 被选择为检查点, 而每个 ACK 包则经过两个中间节点段后被丢弃. 起初, 源节点生成了一个事件包, 并发送给 u_1, u_1 进一步将该事件包沿着路径向基站传递. 当检查点 u_4 收到事件包后, u_4 根据事件包的内容生成一个 ACK 包, 并对 ACK 包内容签名生成一个 MAC 码. 该 ACK 包经过节点 u_3, u_2, u_1 , 最后到达源节点并被丢弃. 这样, 因为 ACK 包的内容是可认证的, 节点 u_3, u_2, u_1 和源节点知道先前的事件包已安全到达检查点 u_4 . 接着 u_3, u_2, u_1 和源节点会继续等待来自下一个检查点(u_6)的 ACK 包.

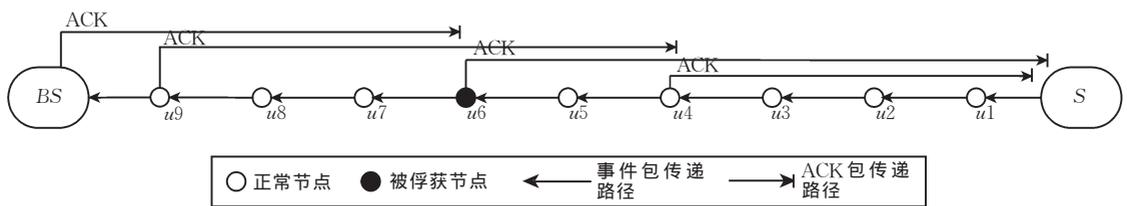


图 3 多跳确认的一个示例(节点 u_9, u_6, u_4 被选为检查点)

下面我们继续考虑如果图 3 中出现了被俘获节点. 假定 u_6 是被俘获节点, 并且恶意丢弃了一个事件包. 则节点 u_5, u_4 无法收到来自下一检查点 u_9 的 ACK 包. 节点 5 生成一个报警包, 指定下游直接邻居节点 u_6 为可疑节点, 然后, 该报警包将经过多跳到达源节点. 当源节点收集足够证据后, 系统进入决策与反击阶段, 例如, 路由协议会减少通过可疑节点的流量.

对于同一个事件包有可能会产生多个报警包, 但这不会显著影响通信量或报警的准确度. 我们可以通过一定的定时机制让节点 u_5 先于节点 u_4 生成

报警包, 并且 u_4 收到 u_5 的报警包后, u_4 不会再生成报警包. 这样, 我们可以避免为一个事件包生成过多的报警包而浪费通信能量. 在模拟实验中, 我们发现由于链路的不稳定性, u_4 仍然可能先于 u_5 产生报警包并将报警包递交至源节点. 但这种情况并不多见, 并且对总体的通信量也影响甚微. 另一方面, 即使针对同一个事件包产生了多个报警包, 源节点也不会因此困惑, 因为它能够轻松地排除误报警. 报警包的 MAC 机制保证了当源节点接收到该包时, 源节点确定先前的事件包至少到达了产生该报警包的节点. 因此在上例中, 即使源节点收到了来自 u_4

和 u_5 的报警包, 源节点知道报警包至少到达了 u_5 , 因此可以排除 u_4 是最后一个“见到”先前事件包的中间节点, 并排除来自 u_4 产生的误报警。

在下面两小节中我们介绍与本文方案密切相关的两个重要问题: 一个是检查点的选择, 另一个是 k -覆盖确认机制。

3.3 随机检查点选择技术

一种简单的方法是在源节点发送第一个事件包前, 事先选择一个固定的检查点名单。这些被事先确定的中间节点在随后的事件包到来后, 担当检查点的角色。但是, 就安全性而言, 这种方法是不可行的, 因为如果检查点是事先确定并且固定的, 检查点会成为敌方俘获的目标。一旦一定数量的检查点被俘获, 则敌方能够轻易丢弃事件包而通过编造 ACK 包的方法来避免其丢包行为不被上游节点检测到。这样, 检查点可能成为导致系统单点失败 (single points of failure) 的重要因素。因此, 我们需要实现一种随机的检查点选择技术使得所有中间节点共享成为俘获目标的风险, 避免单点失败。

本文提出了一种随机检查点选择技术。使用该技术, 源节点会为每一个事件包随机生成一个检查点名单。该技术主要包括两个步骤: 节点初始化和基于随机检查点的确认过程。

(1) 节点初始化。在部署前, 每个节点会加载两个函数: 一个单向 Hash 函数 $F(ID, x)$ 和一个映射函数 $f_p(y)$, 其中, ID 是传感器的 ID, 而 p 是一个预定义的概率值。函数 $f_p(y)$ 具有以下属性: $Range(f_p) \rightarrow \{0, 1\}$, 并且当 $y \in Range(F)$ 时, f_p 的输入值以概率 p 映射到 1, 以概率 $(1-p)$ 映射到 0。

(2) 基于随机检查点的确认过程。源节点为每个事件包生成一个随机数 r , 作为事件包 *Checkpoint_Seed* 字段的值, 然后, 该事件包被一跳一跳向基站传递。当每个中间节点收到这个事件包时, 该节点检查 $f_p(F(ID, r))$ 是否等于 1。如果等于 1, 该节点知道它已被选取为一个检查点, 之后它会生成一个 ACK 包, 并向源节点方向传递。

当一个上游的节点收到了该 ACK 包, 它对该 ACK 包进行两个验证: (1) 验证该 ACK 包是否来自一个真实的检查点。只要通过计算 $f_p(F(ID', r))$ 是否等于 1 即可判断该 ACK 是否来自一个真实的检查点。其中, r 是先前事件包中的 *Checkpoint_Seed* 值, ID' 是 ACK 包中的 *Node_ID* 字段 (即该 ACK 包的生成者)。(2) 验证该 ACK 包的内容是否被篡改, 可以通过 ACK 包上的 MAC 码完成。如果有任

何一个验证没有通过, 则该节点丢弃该 ACK 包; 如果都通过了, 则该节点相信先前的事件包已安全到达下一个检查点, 并接着向上游传递。

当然, 由于随机检查点是随机选择的, 所以可能存在正好没有检查点被选到的情况, 这时整个确认过程成为一种特殊的确认: 端到端的确认, 即由基站直接发出一个 ACK 包经过多跳递交到源节点。这种情况不会对检测方案的正确运行产生影响。

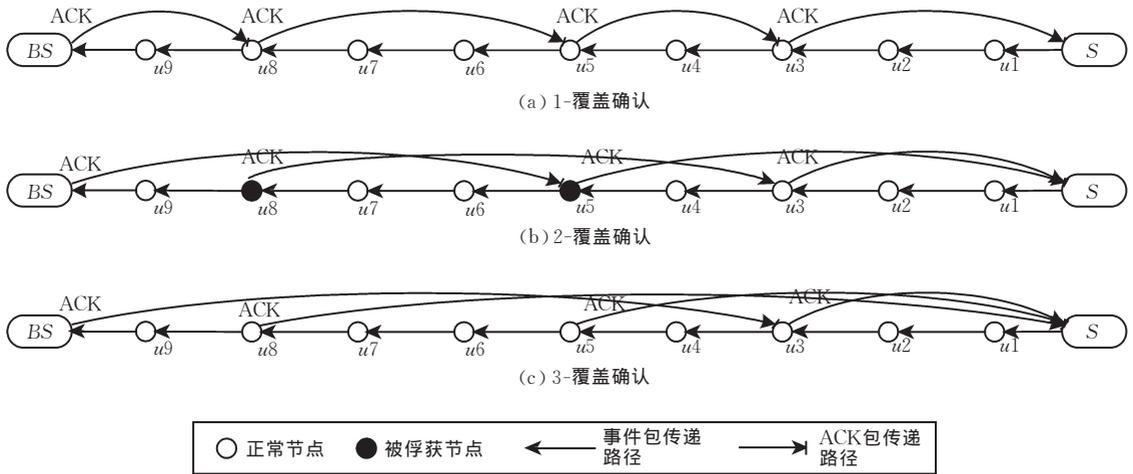
本文提出的随机检查点选择技术具有两方面优势: 一方面, 在一条路径上的所有中间节点享有相同的成为检查点的概率 (这样, 也享有相同的成为俘获目标的风险, 避免单点失败); 另一方面, 我们使用动态的检查点选择, 让敌方无法预知下一次的检查点选择名单。

3.4 k -覆盖的确认

在这一小节中, 我们讨论一个 ACK 包在被某个检查点丢弃之前应该经过多少个段 (一条路径上两个检查点之间的所有中间节点构成一个段)。如果一个 ACK 包经过 k 个段被丢弃, 则我们称是 k -覆盖的确认方案 (k -covered acknowledgement)。

图 4 示例了 3 个 k -覆盖确认的示例。在图中, 我们容易发现如果一个包经过 k 个段后被丢弃, 则每个中间节点对于每个事件包能够收到 k 个 ACK 包, 即每个节点被 k 个 ACK 包所“覆盖”。但是, 靠近基站的段则是例外, 例如图 4(b) 中靠近基站的一个段只被 1 个 ACK 包“覆盖”; 而图 4(c) 中靠近基站的两个段也少于被 3 个 ACK 包“覆盖”。实际上, 靠近基站的段没有必要被 k 个段覆盖, 因为一个中间节点接收到一个来自基站的 ACK 包, 说明事件包已安全到达基站, 因而也没有必要继续等待其它 ACK 包。

参数 k 在系统安全性与通信开销之间提供了一种折中。 k 值越大, 系统安全性越强, 但系统的通信开销也会随之上升。我们可以发现, 当 k 个被俘获节点碰巧被选择为 k 个连续的检查点, 则敌方可以随意丢弃事件包而不被发现。例如, 在图 4(b) 中, 假定节点 u_5 和 u_8 被俘获而且碰巧被选择成为两个连续的检查点。当 u_5 接收到事件包时, 它分别使用自己内存中的密钥信息和 u_8 节点中的密钥信息生成两个 ACK 包。我们假定被俘获节点能够互相合作, 并能够共享他们的密钥信息。这样, 节点 u_4 和其它上游节点收到两个 ACK 包, 认为先前事件包已安全到达检查点 u_5 和 u_8 。实际上, 事件包在 u_5 已被恶意丢弃。所以, 如果我们增加 k 值, 则需要更多的被

图 4 k -覆盖确认的示例

俘获节点被碰巧选择为检查点,从而系统的安全性增强了。另外,值得注意的是,在上例中, u_5 和 u_8 也仅可以有一次机会恶意丢弃而不被发现,因为检查点选择是随机生成的,下一个事件包会有不同的检查点名单。

3.5 攻击分析

这一小节主要讨论了敌方针对本文的检测方案可能产生的一些响应措施以试图避免被检测到。

一个被俘获节点可能更改一个事件包中的 *Checkpoint_Seed* 字段为某一特定的值,以使其自身以及下游其它的被俘获节点被选择为检查点。请注意,在本文方案中,*Checkpoint_Seed* 字段并没有通过任何 MAC 机制进行认证,但这并不会影响本文方案的正常运行,敌方也不会因此受益。例如,在图 3 中,假设节点 u_6 和 u_7 被俘获,并且在 u_6 接收到一个事件包后将其中 *Checkpoint_Seed* 字段更改为某一特定值使得 u_7 成为一个检查点。另外,注意因为 u_5 先于 u_6 接收到先前的事件包,在 u_5 的缓存中仍然保存着正确的 *Checkpoint_Seed* 值。所以,当 u_5 接收到来自 u_7 的 ACK 包时, u_5 能够轻松地验证得出这是一个无效的 ACK 包,而且 u_7 也并非真实的检查点(见 3.3 节),因而 u_5 丢弃该 ACK 包。 u_5 最终将由于收不到足够的 ACK 包而生成一个报警包,检举 u_6 。最后,如我们在 2.1 节中所申明,对事件包内容篡改(即对 *Payload* 及其它字段的撰改)的防范方法不在本文的讨论范围。

对于 ACK 包,一个被俘获节点有可能试图编造虚假的 ACK 包,也可能丢弃来自下游正常节点的 ACK 包。首先,一个被俘获节点是无法编造来自其它正常节点的 ACK 包的,只要该正常节点还没有被俘获。这是由 ACK 包的 MAC 验证码机制所保

证的(见 3.1 节)。第二,丢弃来自正常节点的 ACK 对于被俘获节点来说没有好处。这个行为会阻止上游的邻居节点接收到足够的 ACK 包,因而反而使得被俘获节点更易于被上游节点所检举。

一个被俘获节点可能通过生成一个报警包恶意检举其它节点。实际上,对于一个被俘获但还没有被检测出的恶意节点,确实是不可能阻止它检举无辜的节点的。因此,恶意检举对于敌方是确实可行的,但敌方这种恶意检举行为产生的影响是十分有限的。因为在本文方案中,我们规定任何一个节点只能检举其通信距离内的邻居节点,被俘获恶意节点无法检举网络中的任意节点,恶意检举只是限制在局部。我们仍然能够对网络中发生异常丢包的位置进行跟踪,对可疑节点进行识别,我们会在第 5 节中进一步讨论可疑节点识别的问题。另外,请注意,对于两个节点是否真的在地理位置上相邻,我们可以通过 2.2 节中提出的位置绑定 ID 技术进行验证,这进一步保证了恶意检举的影响是局部化的。

3.6 检测率分析

在这一小节中,我们对本文方案能够达到的检测率进行了理论分析。为了在一定程度上简化问题,我们假定网络运作在一个理想的信道中,即不会因为信道原因导致丢包。这样如果发生丢包,则一定是被俘获节点的恶意丢包。我们会在第 4 节模拟实验中引入信道误码率,进一步考察本文方案在更真实的信道环境中的检测能力。

我们假定在一条传递路径上有 n 个传感器节点,其中 m 个为被俘获节点。为简单化问题,我们假定这 m 个被俘获节点是间隔部署的,即每两个被俘获节点中间有 q 个正常节点($q \geq 0$)。我们采用 k 覆盖的确认机制。 α 是一个百分比,指定了路径上的多

少比例的节点将被选取为检查点。

我们的目标是计算出恶意丢包能够被检测到的概率 $P_{\text{detection}}$, 即等于少于 k 个连续的检查点碰巧是被俘获节点的概率 P_{less_k} . 请注意, 如果有 k 个被俘获节点碰巧被选为检查点, 但它们不是连续的检查点, 即在两个被俘获检查点之间存在正常的检查点, 则这种情况下的恶意丢包仍然是能被本文方案检测到的。

我们可以按如下方式计算检测率 $P_{\text{detection}}$:

$$P_{\text{detection}} = P_{\text{less}_k} = 1 - P_{k_mali}.$$

其中, P_{k_mali} 是至少 k 个被俘获节点被选取为连续的检查点的概率。注意在两个被俘获检查点之间仍然可能有一个或多个被俘获节点, 只不过它们没有被选为检查点。 P_{k_mali} 可以按如下方式计算:

$$P_{k_mali} = \sum_{i=k}^m P_{\text{exact}_i}(i).$$

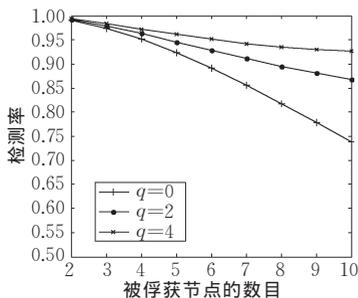
其中, P_{exact_i} 指正好有 i 个被俘获节点被选取为检查点的概率 ($k \leq i \leq m$). 假定在传递路径上, 从第一个被俘获检查点到最后一个被俘获检查点之间共有 j 个被俘获节点 (可能包括部分被俘获节点是非检查点), 则 P_{exact_i} 可以计算为

$$P_{\text{exact}_i}(i) = \sum_{j=i}^m P_{\text{exact}_{i-j}}(i, j).$$

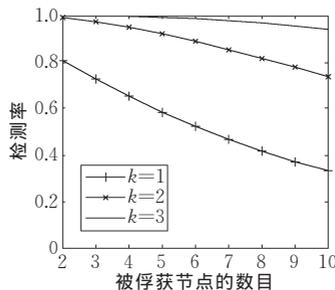
其中, $P_{\text{exact}_{i-j}}$ 表示正好有 i 个被俘获节点被选取为检查点, 并且在第一个被俘获检查点和最后一个被俘获检查点之间正好有 $j-2$ 个被俘获节点的概率。

然后, $P_{\text{exact}_{i-j}}$ 可以计算为

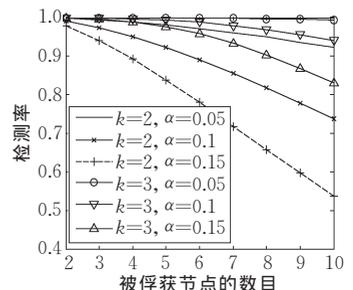
$$P_{\text{exact}_{i-j}}(i, j) = \frac{(m-j+1) \cdot \binom{j-2}{i-2} \cdot \binom{n-m-(j-1) \cdot q}{\alpha \cdot n-i}}{\binom{n}{\alpha \cdot n}}$$



(a) q 和 m 对检测率的影响
(假定 $k=1, \alpha=0.1, n=100$)



(b) k 和 m 对检测率的影响
(假定 $q=0, \alpha=0.1, n=100$)



(c) α 和 m 对检测率的影响
(假定 $q=0, n=100$)

图 5 q, k, α 和 m 对检测率的影响

通过上述分析, 我们可以获取以下 3 个本文检测方案所具有的特性: (1) 为避免被检测到, 敌方更倾向于俘获一条传递路径上的连续节点, 而不是分

最后, 总的检测率 $P_{\text{detection}}$ 可以按照如下公式计算:

$$P_{\text{detection}} = 1 -$$

$$\sum_{i=k}^m \sum_{j=i}^m \frac{(m-j+1) \cdot \binom{j-2}{i-2} \cdot \binom{n-m-(j-1) \cdot q}{\alpha \cdot n-i}}{\binom{n}{\alpha \cdot n}}.$$

我们根据上述理论分析的结果考察了参数 q, k, α 和 m 对检测率的影响, 见图 5. 我们假定传递路径上有 100 个中间节点 ($n=100$). 首先, 我们考察了被俘获节点的部署方式 (q 值) 对检测率的影响. 图 5(a) 显示与 $q=2, q=4$ 的曲线相比, $q=0$ 的曲线可以导致比较低的检测率, 这说明敌方更喜欢俘获在一条传递路径上相邻的节点, 而不是分散的节点, 这样可以有利于避免被检测到. 因此, 为了分析本文方案在最坏的情况下的检测性能, 在图 5(b), (c) 中, 我们都假定被俘获节点是连续分布的, 即 $q=0$. 下面, 我们检查参数 k 对检测率的影响, 如图 5(b) 所示. 当 $k \geq 2$ 时, 本文方案显示出较高的检测率, 即使当 10 个节点被俘获后 (10% 的节点被俘获, 通常认为是比较严重的入侵), 检测率仍然高于 75%. 但是, 当 $k=1$ 时, 检测率相当低. 这也很容易理解, 因为当 $k=1$ 时, 只要有任一被俘获节点碰巧被选为检查点时, 敌方即可随意丢包而不被检测到. 最后, 图 5(c) 显示了 α 对检测率的影响. 我们发现当 k 与 m 固定时, 较小的 α 值会带来较大的检测率, 这是因为选取更少的检查点, 也可以避免选取到被俘获节点的概率, 因而检测率上升了. 但是, 在真实的通信条件下, 过小的 α 值, 也可能导致延长的检测时间以及由于信道丢包误报警等. 我们会在未来的工作中进一步研究检查点的部署与检测能力的关系。

散的节点. (2) $k \geq 2$ 是本文方案的一个最低安全要求. (3) 在理想的链路状态下, 选择较少的检查点, 有利于提高检测率。

4 模拟实验

在这一节中,我们通过模拟实验对本文的检测方案进行深入地评估.在模拟实验中,我们采用了更为真实的场景.在这个场景中,我们假定恶劣的信道条件同样会导致丢包.我们的实验评估主要集中于两方面:检测的准确度和通信开销.

实验场景假定 400 个传感器节点均匀分布在 $2000 \times 2000 \text{m}^2$ 的场地,一个固定的基站和一个事件源分别在这个正方形场地的两边.为评估系统的平均性能,我们假定事件源生成 500 个事件包,并且整个实验重复进行了 10 次.节点之间的通信速率是 19.2Kbps.我们在每个节点上实现了一个类似 PSFQ^[7] 的传输层重传机制,以应对恶劣的信道条件.我们让信道误码率在 0~15% 之间变化,考察恶劣信道条件对本文方案的影响.通常大于 10% 的信道误码率就被认为是十分恶劣的信道条件.为了评估本文方案在最坏条件下的检测能力,我们假定敌方俘获了一条传递路径上的若干连续节点,即 $q=0$ (基于 3.6 节的理论分析结果).

我们提出以下 3 个评估参数:

(1) 检测率,即检测概率.它是检测到的恶意丢包数量与全部恶意丢包(包括未检测到的)数量的比值.

(2) 误警率,即 false positive rate.它是检测到

的非恶意丢包(恶劣链路导致丢包)数量与全部检测到的丢包数量(包括链路原因和恶意丢包)的比值.

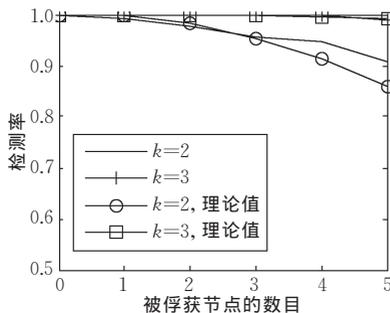
(3) 相对通信开销.它是采用了本文检测方案后系统总的通信开销与没有采用本文检测方案的通信开销的比值.

其中,前两个参数集中于评估检测方案的准确度,而相对通信开销试图将本文方案与其它防御方案做比较,例如多路径传递.

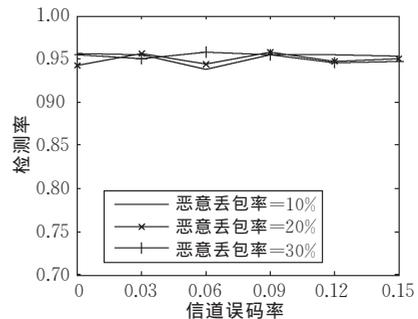
4.1 检测准确度

在这一小节中,我们研究信道误码率(channel error rate)、恶意丢包率(malicious dropping rate)、传输层重传机制对检测准确度的影响.

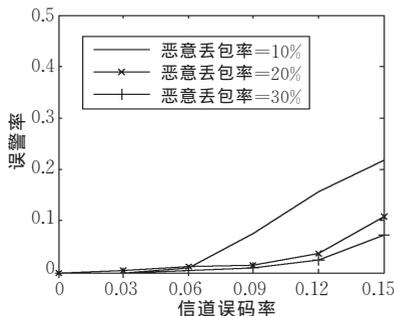
我们的第 1 个模拟实验显示了 k 值和被俘获节点数量 m 对检测率的影响.图 6(a) 显示出检测率随着 k 值的增加而增加,而随着被俘获节点数量 m 的增加而减小.并且我们容易发现两条检测率的实验结果曲线基本服从理论分析的结果.实验结果基本如我们所预料, k 值的增加使得足够的被俘获节点被选取为检查点的概率变小,因而检测率上升.给定 $k=2$,即使当传递路径上 25% 的节点(5 个节点)被俘获了,检测率仍然能够高于 90%.另外,需要注意的一点是,我们的理论分析是假定在理想的信道环境中的,但如图 6(a) 所示,理论分析结果和模拟实验结果差别并不大.这暗示着信道误码率对检测率的影响并不大,我们会在后面的实验中进一步考察信道误码率的影响.



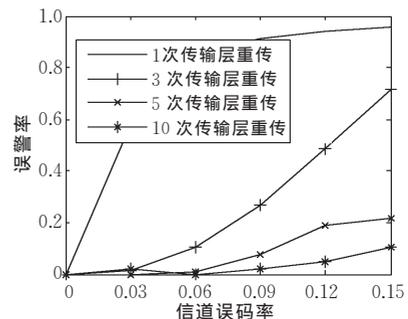
(a) m 对检测率的影响(假定信道误码率=10%)



(b) 信道误码率对检测率的影响(假定 $k=2, m=3$)



(c) 信道误码率对误警率的影响(假定 $k=2, m=3, \text{重传次数}=5$)



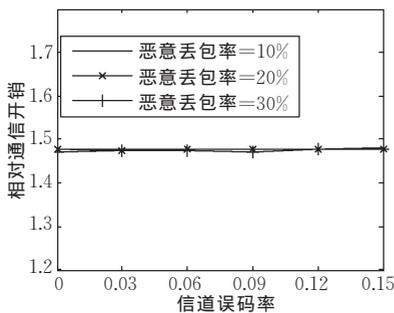
(d) 重传机制对误警率的影响(假定 $k=2, m=3, \text{恶意丢包率}=10\%$)

图 6 检测准确度

我们的第 2 个实验考察了信道误码率和恶意丢包率对检测率的影响. 图 6(b)的结果显示了信道误码率与恶意丢包率对检测率都没有明显的影响. 我们认为, 增加的信道误码率会导致更多由于信道原因的丢包, 但它并不能阻止被丢弃的包被检测出来.

我们的第 3 个实验测试了信道误码率和恶意丢包率对误警率的影响. 如图 6(c) 所示, 信道误码率与恶意丢包率都对误警率有显著的影响. 增加的信道误码率会导致更多的包由于信道原因而丢弃, 但是对丢包的原因进行明确的区分是十分困难的, 因而误警率随信道误码率上升而上升. 但是, 我们认为, 只要一个被俘获节点在一定的信道条件下比一个正常节点丢弃更多的包, 那么这种恶意丢包的攻击总是能被检测到的. 例如, 我们假定恶意丢包率为 20% 而信道误码率为 15% (通常被认为是十分恶劣的信道条件), 则此时误警率为约 10% (如图 6(c)), 检测率大于 95% (如图 6(b)).

我们的第 4 个模拟实验显示了传输层重传机制能够有效地减少由于信道原因导致的丢包, 从而也降低了误警率. 如图 6(d) 所示, 即使当信道误码率为 15%, 恶意丢包率 10% 时, 如果我们允许 10 次传输层重传, 则能够有效减少由于信道原因导致的丢



(a) 信道误码率与恶意丢包率的影响

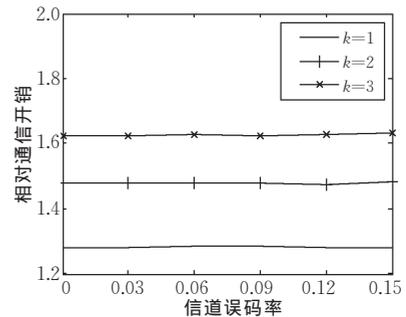
包, 从而误警率仍然能够保持在 10% 以下.

4.2 通信开销

我们试图使用相对通信开销来将本文的方案与其它抵抗选择传递攻击的方案进行比较. 实验结果基于图 2 包格式的定义.

图 7(a) 调查了信道误码率和恶意丢包率对相对通信开销的影响. 增加的信道误码率会导致更多的包需要通过重传完成传输, 这必然导致增加的总通信开销, 本文的检测机制所导致的通信开销也是按比例相应增加, 所以相对通信开销的变化很小. 同样, 敌方恶意丢弃包的多少也对相对通信开销的影响很小.

如图 7(b) 所示, k 值成为一个影响相对通信开销的重要参数. 这一点是容易理解的, k 值决定了路径上的每个节点会被多少 ACK 包所“覆盖”, 从而直接影响了相对通信开销的大小. 从图 7 中, 我们可以发现不论 $k=3$ 或 $k=2$, 相对通信开销的大小都是保持在 1.7 以下的. 显然, 就通信开销这一点上, 本文的方案与多路径传递方案^[1] 相比具有一定的优势. 在多路径传递方案中, 通信量会随着路径数量的增加而成比例增加, 即使只有两条路径存在, 通信开销也至少是单路径传递的 2 倍以上.



(b) k 对通信开销的影响

图 7 通信开销

5 结 论

本文提出了一种简单有效的选择传递攻击检测方案. 许多传统的传感器网络入侵检测方案一般是基于基站或者基于其它的中央式检测机构. 而传统的入侵检测方案不同, 本文的检测方案中, 每个源节点具有收集来自中间节点报警信息的能力. 这种能力使得即使基站在攻击中无法获取足够信息时, 源节点仍然有可能进行检测并做出决策. 选择传递攻击确实是一个十分具有挑战性的研究问题. 在未来的工作中, 我们期望深入研究其它几种抵抗选择

传递攻击的方法, 进一步完善系统的总体防御能力.

参 考 文 献

- 1 Karlof C., Wagner D.. Secure routing in wireless sensor networks: Attacks and countermeasures. In: Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, 2003, 113~127
- 2 Intanagonwivat C., Govindan R., Estrin D.. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: Proceedings of the ACM MobiCom 2000, Boston, USA, 2000, 56~67
- 3 Karp B., Kung H. T.. GPSR: Greedy perimeter stateless rou-

- ting for wireless networks. In: Proceedings of the ACM Mobicom 2000, Boston, USA, 2000, 243~254
- 4 Yu B., Xiao B.. Detecting selective forwarding attacks in wireless sensor networks. In: Proceedings of the 2nd International Workshop on Security in Systems and Networks (IPDPS 2006 Workshop), Greece, 2006
 - 5 Adrian P., Robert S., Victor W., David C., Doug T.. SPINS: Security protocols for sensor networks. In: Proceedings of the ACM Mobicom 2001, Rome, Italy, 2001, 189~199
 - 6 Yang H., Ye F., Yuan Y., Lu S., Arbaugh W.. Toward resilient security in wireless sensor networks. In: Proceedings of the ACM MobiHoc 2005, Cologne, Germany, 2005, 34~45
 - 7 Wan C. Y., Campbell A. T., Krishnamurthy L.. PSFQ: A

- reliable transport protocol for wireless sensor networks. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, USA, 2002, 1~11
- 8 Zhu S., Setia S., Jajodia S., Peng N.. An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, USA, 2004, 259~271
 - 9 Lamport L.. Constructing digital signatures from one-way function. SRI International; Technical Report SRI-CSL-98, SRI International, 1979



YU Bo born in 1978, Ph. D. candidate. His research focus on wireless sensor networks.

YANG Min, born in 1979, Ph. D. candidate. His research interests focus on wireless sensor networks.

WANG Zhi, born in 1977, Ph. D. candidate. His research interests focus on grid computing.

GAO Chuan-Shan, born in 1942, professor, Ph. D. supervisor. His research interests computer network and distributed system.

Background

Wireless Sensor Networks (WSNs) are ideal candidates for monitoring environments in a wide variety of applications such as military surveillance and forest fire monitoring. In such a network, a large number of sensor nodes are deployed over a vast terrain to detect events of interest (e. g., enemy vehicles, outbreaks of forest fires), and to deliver data reports to the base station over multi-hop wireless paths. The node-patterned deployment of WSNs, however, can be focused of certain types of malicious attack. One such strategy is the selective forwarding attack, first proposed by Karlof. In such attacks, a malicious node selectively drops sensitive packets, for example, a packet reporting the enemy tank movements. Selective forwarding attacks are typically most effective when the attacking nodes are explicitly included on the path of a data flow. They can corrupt a number of existing routing protocols such as TinyOS beaconing, Directed Diffusion, GPSR, GEAR, and clustered based protocols, especially when they are used in combination with other attacks such as wormhole and sinkhole attacks. The adversary may incur abnormal packet loss in two ways, from inside the network via maliciously dropping packets going through compromised nodes or from outside the network by jamming the communication channels between uncompromised nodes. Usually, adversaries prefer inside attacks because they put

the adversary in a position to know more about passing packets, thereby enabling them to selectively drop sensitive packets. In this paper, we also mainly focus on selective forwarding attacks from inside compromised nodes.

One possible approach that can be used to decrease the impact of selective forwarding is to use a multipath forwarding technique, which is based on packet delivery redundancy. However, multipath forwarding suffers from several drawbacks. First, communication overheads increase dramatically as the number of paths increase. Second, multiple paths ultimately join up in the area neighboring the base station, so if nodes around the base stations are compromised, selective forwarding is still applicable. Finally, the multipath forwarding shows poor security resilience. To compromise the system, an adversary merely needs to ensure the presence of one compromised node in each path.

In this paper, the authors propose CHEMAS (CHECKpoint-based Multi-hop Acknowledgement Scheme), a lightweight security scheme that detects selective forwarding attacks by using a checkpoint-based acknowledgement technique. Usually, a security system consists of detection and response, while this paper mainly focuses on the detection of selective forwarding attacks. More work on the response aspect is still required as the future work.