无线自组织网络下抵抗内部节点丢弃报文攻击的安全通信模型

张中科 汪 芸

(东南大学计算机科学与工程学院教育部计算机网络和信息集成重点实验室 南京 210096)

摘 要 无线自组织网络的报文传输是依靠网络中的节点彼此多跳接力传输,当网络中有节点被俘获以后,就会成为内部攻击者,并在报文的传输过程中发起丢弃报文攻击,严重降低网络性能.现有网络协议栈中传输层和网络层协议难以检测和防范在网络层发起丢弃报文攻击的节点.文中提出一种在网络层抵抗内部节点丢弃报文攻击的通信模型,它包括通信链路状态实时分析协议和分布式的节点类型判定算法两个部分.通信链路状态实时分析协议利用节点对间逐段生成的路径环路,通过引入报文成组应答机制使得节点能实时地获取其邻居节点的报文转发状态;在通信链路状态实时分析协议基础之上,节点根据相关数学模型能有效地分析邻居节点行为是否异常,并对网络中的节点进行分类,最终将恶意节点从网络中隔离出去.仿真实验结果表明,文中算法在恶意节点的检测率和误检率方面性能表现良好,能有效地抵抗来自网络内部节点的丢弃报文攻击.

关键词 内部攻击;报文丢弃攻击;恶意节点检测;安全通信模型;无线自组织网络

中图法分类号 TP393 **DOI 号**: 10.3724/SP.J.1016.2010.02003

A Secure Communication Model for Defending Against Insider Packet Dropping Attacks

ZHANG Zhong-Ke WANG Yun

(*Key Laboratory of CNII of Ministry of Education, School of Computer Science & Engineering, Southeast University, Nanjing* 210096)

Abstract In wireless ad hoc networks, packets are delivered by multi-hop relay among nodes. When some nodes are captured, they turn out to be inside attackers, then drop data packets arbitrarily. Such kind of attack, called packet dropping attack, will dramatically degrade network performance. However, neither transport layer nor network layer protocols of existing network protocol stacks are able to defend against it efficiently. Hence, this paper presents a new secure communication model defending against packet dropping attack from inside attacker in the network layer, including Real-time Link Status Analysis protocol (RLSA) and Distributed Node Classification algorithm (DNC). RLSA took advantage of segment-based multiple paths between a pair of source and destination, as well as group acknowledgement mechanism for packets, to obtain packet forwarding status of neighbors in real time. Based on RLSA and related analytical model, DNC identifies abnormal behavior of nodes, and then classifies nodes into normal and malicious types. Simulation results show the secure communication model performs well regarding malicious node detection rate and false alarm rate.

Keywords insider attack; packet dropping attack; malicious node detection; secure communication model; wireless ad hoc networks

收稿日期:2010-08-22.本课题得到国家自然科学基金(60973122)和国家"九七三"重点基础研究发展规划项目基金(2009CB320705)资助.**张中科**,1980 年生,男,博士研究生,主要研究方向为无线自组织网络安全.E-mail: zhangzhongke@seu.edu.cn. **汪 芸**(通信作者),1967 年生,女,博士,教授,博士生导师,主要研究领域为分布式计算、容错理论和无线自组织网络.E-mail: yunwang@seu.edu.cn.

1 引 言

无线自组织网络适合在没有基础设施的情况下构建网络环境,网络中的节点可以通过自组织方式临时自主组网,在环境监测、救灾以及人们的日常生活等方面具有广泛的应用前景.但是由于无线自组织网络中缺少中心管理者,网络通常部署在开放的环境中,无线信号暴露在空间,使得无线网络中的节点易于被俘获,节点间的通信容易被窃听、伪造和重放,从而导致无线自组织网络的应用受困于各种类型的网络攻击.

在无线自组织网络中,数据报文通过网络中的节点彼此接力多跳传输,当网络中的合法节点被攻击者俘获以后将成为恶意节点.在数据报文传输过程中,恶意节点可以故意随机丢弃部分经过的数据报文来对网络通信实施破坏,这就是内部节点丢弃报文攻击,它导致网络吞吐率下降、报文重传率增高,严重时造成网络报文传输无法进行.然而现有的TCP/IP协议栈无法应对这种在网络层发起的攻击,加上无线自组织网络中节点性质的不确定性,使得其难以检测和防范,严重影响到了无线自组织网络的通信性能和实际应用.文献[1]描述了一种无线自组织网络中针对流媒体的丢弃报文攻击方法,并通过仿真实验发现该攻击最终导致流媒体无法正常播放.

解决内部节点在网络层发起的丢弃报文攻击的关键在于找出丢弃报文的恶意节点,同时需要兼顾到如下几个方面:(1)检测算法必须遵从无线自组织网络的自治性,仅依靠网络中节点间的彼此协作检测出恶意节点;(2)算法必须能够及时准确地检测出网络中的恶意节点,同时保持较低的误判率;(3)必须保证算法的效率,算法的计算和存储负荷不应过大;(4)算法必须具有通用性,能够与不同的路由协议有机地结合,且能够应对较强的攻击模型.但是现有的研究成果有的检测效率和通用性不够理想,有的难以应对强的攻击模型,有的存在片面性,不能将节点的行为监测和节点状态分析有机地结合起来.

为了克服现有研究成果的不足,本文提出了一种无线自组织网络下分布自治式抵抗内部节点丢弃报文攻击的通信模型,它包括通信链路状态实时分析协议和分布式节点类型判定算法两个部分.通信链路状态实时分析协议充分利用了无线自组织网络存在冗余路径的特点,在分段多路径中主路径和备份路径形成的环路基础之上,通过网络层的报文成组转发和ACK机制来监测邻居节点的报文转发行为.分布式节点类型判定算法根据已获取的邻居节点的行为记录,依据相关数学模型,通过统计分析和局部选举的方法来及时准确地分类网络中的节点,最终将丢弃报文的恶意节点从网络中隔离出去,从而构建安全健壮的网络通信环境.与已有方法相比,本文提出的解决方案,可架构在任何路由协议基础之上,遵循了无线自组织网络节点分布自治的原则,具有实时性和简单有效等特点,且能够应对强攻击模型.

本文的主要贡献有:

(1)设计了基于报文成组转发机制的通信链路状态检测算法,该算法利用分段多路径路由协议生成的主路径和备份路径,通过发送链路控制报文,收集邻居节点的报文转发状态,分析邻居节点在报文转发过程中是否存在恶意行为.

(2)基于通信链路检测协议的检测结果,建立了适合于自组织网络特性的简单有效的评判节点行为的数学模型,设计了选举可信节点和识别隔离恶意节点的算法.

(3)实验仿真分析了算法在检测率和误检率两方面的性能表现,实验结果表明本文算法具有高恶意节点检测率和低误检率.

本文第2节介绍研究现状和相关技术;第3节给出网络模型和攻击模型;第4节设计通信链路状态实时分析协议,描述当网络中出现组报文丢弃后,不同场景下节点对其邻居节点报文转发行为的分析检测过程;第5节建立相关数学模型,设计分布式节点类型判定算法;第6节通过仿真实验分析本文算法的性能指标;最后在第7节总结全文并给出下一步工作的思路.

2 相关工作

(1)基于局部管理(local monitoring)的报文转发监测方法

文献[2]提出了Watchdog算法,节点在混杂模式下工作,当节点把报文转发给下一跳节点后,利用无线信号的暴露在空中的特性来监听下一跳节点有没有继续转发该报文.文献[3]采用了类似Watchdog的邻居检测系统(NWS)来获取邻居节点的报文状况,但是局部管理的方法增加了节点能量耗费,降低

了网络带宽使用效率. 文献[4]指出恶意攻击者可以通过控制通信范围，将报文传送给不存在的下一跳节点，或者故意造成无线通信信号冲突等方式来降低局部管理监测的准确率. 因此局部管理方法在效率和准确性上均存在问题.

(2) 基于报文应答的丢包检测方法

文献[5]提出了一种基于 ACK 应答机制的可靠报文传输模型，当节点发送报文时将设置计时器，等待下两跳邻居节点发送过来确认报文. 若收不到下两跳邻居节发送的 ACK 报文，就认为下一跳邻居节点和下两跳邻居节点之间的链路存在问题. 该算法仅适用于防范较弱攻击. REAct 算法[6] 中源节点通过 Bloom Filter 和随机二分查找算法来定位恶意丢包节点. ODSBR 协议[7] 则通过 ACK 报文和折半查找式审计报文来判定路径上的恶意节点. 该两种协议都需要源路由的支持，且难以应对合谋攻击. AAI 协议[8] 有两种形式，PAAI-1 和 PAAI-2. 在 PAAI-1 中，中间节点只按概率应答部分数据包，在 PAAI-2 中，对于源节点发送出来的所有报文，只有部分中间节点向发送源节点应答报文. 但是该方法检测实时性差，且资源消耗难以适应无线网络设备能力有限的特点.

(3) 基于信任模型的节点评估方法

文献[9-10]通过发送 ACK 和 NACK 报文的方法来使源节点和沿途节点获得自己下游邻居节点的报文转发情况. 文中提出了一种定量的信任模型，基于节点的报文转发历史情况计算节点的“可信任”值，再根据该值选择一条最可信的路径. 该方法需要源路由的支持，每个报文都需要 ACK 确认，且难以应对恶意节点合谋攻击. 文献[11-12]认为信任是对不确定性的一种度量，根据节点以往转发报文的成功率来估计未来报文转发成功率，定量测度信任值，但是没有提出有效获取链路上节点转发报文情况的方法.

(4) 基于统计分析的节点评估方法

文献[13]提出了一种基于统计分析检测恶意节点内部节点的模型，该方法用节点丢包率、报文的发送率和延迟等数据来建立节点的属性向量，估计出属性向量总体的均值和方差. 通过计算各个节点样本到均值向量的马氏距离来检测异常节点即恶意攻击者. 但是没有给出如何有效地获得节点的行为属性参数的方法. 文献[14]提出一种异质网络下的丢弃报文节点检测方法，该方法中设备能力更强的簇头节点通过报文加密手段来获得普通节点的报文转发情况，通过 SPRT 来判定簇内的恶意节点，但是该方法需要异质网络结构的支持，且数据报文加密传输增加了节点运算处理负荷. 文献[15]通过对数据报文进行加密填充处理来使恶意节点无法识别发送报文的源节点和中间节点，然后依据相关模型来检测出链路上的恶意丢弃和修改报文的节点. 文献[16]认为网络中的链路往往出现在不同路径上，而这些路径上的报文传递率有好有坏，借助于“序贯概率比检测”，可以确定出传感器实际出错链路所在的位置. 上述两种方法计算量都较大，且只适用于以汇聚(sink)节点为通信中心的网络模式.

3　系统模型

3.1　问题描述

在无线网络中数据链路层维护着节点同其邻居节点间的通信链路，并利用差错检测、流量控制、报文确认和重传机制来保证数据帧的可靠传输. 虽然数据链路层可以提供报文在节点和其邻居节点间的可靠传输，但是并不能保证报文端到端的正确可靠传输，因为即使报文可靠地传输到下一跳节点，也可能因为种种原因报文在下一跳节点上丢失，比如内存拷贝错误、缓冲区溢出甚至恶意节点的故意丢弃等.

传输层协议维护着源节点和目的节点间的通信连接服务功能，TCP/IP 协议栈中传输层协议有 TCP 和 UDP 两种类型. TCP 是一种面向连接的、可靠的传输层通信协议，但是当 TCP 发现报文传输出现异常时，并不能确定报文在何处出现故障，即使重新启动路由发现过程，恶意节点仍可能再次出现在报文传输路径上. UDP 提供一种简单的不可靠信息传送服务，它无法应对丢弃报文攻击，虽然基于 UDP 的应用通常并不要求正确传输所有数据报文，但是当丢包率达到一定程度后，上层应用仍会受到极大破坏.

TCP/IP 协议栈在网络层采用无连接的数据报文存储转发机制，虽然可以有效提高报文转发的效率，但是没有提供可靠的报文转发机制. 而传输层和数据链路层提供的报文转发可靠机制并不能解决攻击者在网络层实施的丢弃报文攻击. 相对于有线网络在网络层有专用的设备来负责报文转发，易于管理和维护，无线自组织网络则依靠节点之间多跳接力传输报文，因此恶意节点的丢包攻击产生的危害更加严重.

本文着力解决的问题是针对无线自组织网络中的按需路由机制，设计一种新的通信模型，可以有效地防止由内部节点发起的丢弃报文攻击.

3.2 攻击模型

本文旨在分析和抵抗较强的攻击模型，即网络中可能存在一个或者多个合谋的恶意节点，这些恶意节点参与路由发现过程，但是在数据传输过程中，有选择地丢弃部分接收到的数据报文. 而且，恶意节点能识别分析网络中的控制报文，能根据有利于自己的原则选择传递、丢弃或者伪造.

此外，本文将节点的恶意行为分为以下类型：

(1) 当节点发现邻居节点声称转发的报文个数与实际转发的报文个数不一致时，它认为该邻居节点实施了 I 类恶意行为；

(2) 当节点发现邻居节点声称转发的报文个数与其下游节点声称转发的报文个数不一致时，它认为该邻居节点实施了 II 类恶意行为.

(3) 当节点发现邻居节点发送了两个完全不同的控制报文时，它认为该邻居节点的行为是绝对恶意行为，并认定该节点为恶意节点.

3.3 系统假设

假设任意节点对之间生成了分段多路径(如图 1 分段多路径路由示意图所示)，各分段的段首尾节点为可靠节点，称为关键节点. 关键节点知道所连接段内所有节点的 ID. 假设除了主路径之外，总是存在一条安全的备份路径.

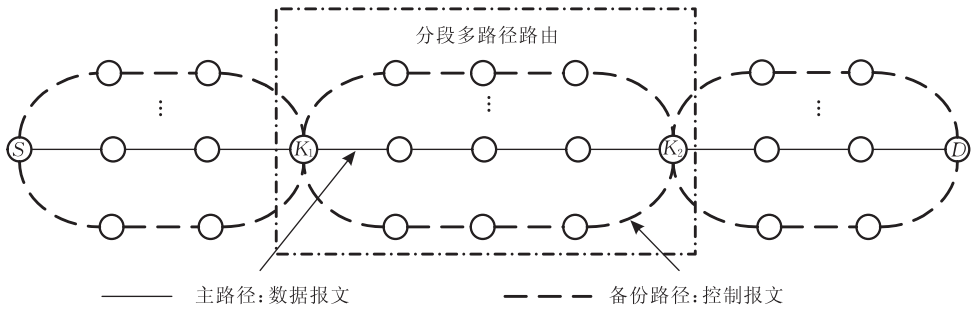


图 1 分段多路径路由示意图

网络中的节点分为普通节点、恶意节点和关键节点. 所有非恶意节点预置了用于签名和认证的密钥对和密钥证书. 如果节点的平均丢包率为 P_d，平均路径长度为 l，则路径的丢包率为 $P_d^{\text{path}}=1-(1-P_d)^l$，若 $l=10$，$P_d=0.1$，$P_d^{\text{path}}=0.65$，因此本文假设正常运行的网络中非恶意节点的丢包率小于 0.1.

4 通信链路状态实时分析协议

4.1 概 述

通信链路状态实时分析协议采用报文组控制方式，当源节点发送的数据报文经过网络层协议栈时，相应的报文组信息会填入报文头部. 如果路径上不存在恶意节点时，路径上任一节点转发完成一个组内的所有报文后，等待并接收目标节点的组确认报文；当目标节点收到该组的所有报文后，将会向源节点发送组确认报文. 但是如果路径上有恶意节点故意丢弃报文时，目标节点将无法收到该组内的所有报文，也就无法向源节点发送确认报文. 路径上的任一节点都配有报文计时器，这些计时器超时将报告网络中的非正常情况，此时触发通信链路实时分析协议的进一步执行. 通过在路径上各节点间发送、接收和分析链路状态检测报文，即 ACK 和 NACK 报文，检测链路状态，每个节点形成相应的邻居节点状态表. 根据所收集到的节点状态信息，检测邻居节点在报文转发过程中的恶意行为.

4.2 网络层报文组织

网络层将传输报文编成组有很多优点：(1) 相对于每个报文确认，成组报文确认可以有效地减少网络中确认报文的数量，提高网络传输效率；(2) 通过对每个报文附上组编号和组内序号，可以使路径上的每个节点动态感知已收发了组中报文数量和缺少的报文数量. 发送的报文组的头部包含 (GID, $GVOL$, PID) 信息，其中 GID 是顺序递增的组编号，$GVOL$ 标示该组内的报文数量，PID 是该报文顺序递增的组内编号.

在网络通信中，下游节点收到的组报文的报文个数小于或等于上游节点收到的报文个数. 如果下游节点收到的报文个数小于上游节点收到的报文数，意味着在它的上游存在一个或者多个节点丢弃了报文.

4.3 链路检测控制报文

链路检测控制报文有两种类型：ACK 报文和

NACK 报文. ACK 报文格式为 $ACK(GID, NodeID, HopToSrc)$，其中 $NodeID$ 为发送 ACK 报文的节点编号，$HopToSrc$ 为 $NodeID$ 节点到源节点的跳数，ACK 报文用来表示 $NodeID$ 节点已转发了编号为 GID 报文组中的所有报文. NACK 报文格式为 $NACK(GID, NodeID, HopToSrc, DropNum)$，用来反馈 $NodeID$ 节点尚未收到 GID 报文组中 $DropNum$ 个报文.

为了防止恶意节点伪造和篡改，需要采用数据签名来保证数据的完整性和数据起源的可认证性. 当节点生成链路检测控制报文向上下游的关键节点传送时，处于同一段内的普通节点依次附上各自签名，以防止段内多个恶意节点合谋伪造链路控制报文，格式为 $(N_i, N_{i-1}, \cdots, r, (N)ACK, Signatrue_{N_i}, Signatrue_{N_{i-1}}, \cdots)$，其中 $N_i, N_{i-1}, \cdots$ 分别为沿途节点 ID，r 为随机数可用来防止重放攻击，$Signatrue_{N_i}$，$Signatrue_{N_{i-1}}$，… 分别为沿途节点对 $(N_i, r, (N)ACK)$ 的签名. 当该报文传送到段首或段尾的关键节点时，关键节点验证报文的传输路径及链路上各节点的签名是否正确，若不正确则通告传输该报文的沿途节点“报文有误”，否则用自己的签名替代上述所有的签名继续传递，格式为 $(r, (N)ACK, Signatrue_{K_i})$.

此后，所有接收到该链路检测控制报文的节点只需验证关键节点对该报文的签名是否正确即可判定报文的完整性和数据起源的可靠性. 当链路检测控制报文在不同的关键节点间传送时，新的关键节点验证签名无误后，将用自己的签名替换前一个关键节点的签名，这样普通节点只需要验证自己所在段的关键节点的签名是否正确即可判定报文的真实性.

4.4　链路状态检测过程

节点配有两个计时器，即 $Timer_1$ 和 $Timer_2$. $Timer_1$ 用于等待目标节点 ACK 报文，其时长正比于节点到目标节点的跳数；$Timer_2$ 在转发下游段节点转发过来的 ACK 报文后设置，用于等待上游节点 NACK 报文，其时长正比于节点到发送 ACK 报文节点之间的跳数，该信息可从 ACK 报文中的 $HopToSrc$ 推算出来. 当 $Timer_1$ 发生超时，触发链路状态检测过程，不同类型的节点执行不同的动作，算法如表 1 所示.

表 1　节点转发报文算法

action of Normal Node	action of Normal Node (continued)

Totally forward batch packets:
　set $Timer_1$ for this batch on;
$Timer_1$ times out:
　send ACK to upstream node;
Received ACK:
　cancel $Timer_1$;
　if(ACK from $Dst \parallel HopToSrc_{pkt} < RXACK.ACK\_HopToSrc$)
　　return;
　record RXACK($GID, ACK\_HopToSrc$);
　forward ACK to upstream node;
　if($HopToSrc_{pkt} < HopToSrc_{self}$)
　　set $Timer_2$ for this batch on;
$Timer_2$ times out:
　　send NACK to downstream node;
Received NACK :
　cancel $Timer_1$ and $Timer_2$ for this batch;
　if($HopToSrc_{pkt} > RXNACK.ACK\_HopToSrc$)
　　return;
	if($DroNum_{pkt} = DroNum_{self}$)
	　record RXNACK($GID, ACK\_HopToSrc$);
	　forward NACK to downstream node;
	else if($DroNum_{pkt} < DroNum_{self}$ && $HopToSrc_{pkt} < HopToSrc_{self}$)
	　set $Timer_2$ for this batch on;
	additional action of Key Node
	Received ACK from node in my segment:
	　verify and re-signatured ACK packets;
	　forward ACK to downstream key node by both path;
	Received NACK from node in my segment:
	　verify and re-signatured NACK packets;
	　forward NACK to upstream key node by both path;
	Received ACK from upstream key node:
	　verify and re-signatured ACK packets;
	　forward ACK to upstream node
	Received NACK from downstream key node:
	　verify and re-signatured ACK packets;
	　forward NACK to downstream node

Note: RXACK records information of ACK from the most remote downstream node	
Note: RXNACK records information of NACK from the most remote upstream node	

(1) 普通节点

普通节点转发其它节点的链路控制报文，且向其它节点诚实通告自己的报文转发情况. 当 $Timer_1$ 发生超时，仍没有收到来自目标节点的关于报文组的 ACK 报文时，普通节点向其上游节点传送 ACK 报文. 当它收到下游关键节点转发过来的段内上游节点生成的 ACK 报文，它将设置 $Timer_2$ 计时器等待上游节点的 NACK 报文. 当 $Timer_2$ 发生超时，它将向下游节点传送 NACK 报文通告下游节点自己收到和转发的报文数量.

如果普通节点收到上游节点 NACK 报文中的 $DropNum$ 等于自己的未收到报文数时，它只是简单向下游节点转发 NACK 报文. 当它收到下游关键节点转发过来的段内上游节点 NACK 报文中的 $DropNum$ 小于自己未收到报文数时，设置 $Timer_2$ 计时器等待上游节点的 NACK 报文. 当 $Timer_2$ 发

生超时，它用自己的 *DropNum* 生成 NACK 报文通告下游节点.

(2) 关键节点

关键节点是可靠节点，位于各个分段的段首和段尾，相邻关键节点之间的备份路径同主路径形成一个闭合环路. 关键节点除了像普通节点一样诚实通告和转发链路控制报文外，还负责链路控制报文的验证及其在上下游段之间的传递.

如果关键节点收到下游关键节点发送来的 ACK 报文，说明在它和紧邻的下游节点之间没有报文被丢弃，将 ACK 报文向上游节点继续传递，直至报文最终到达源节点，沿途各节点通过该报文可以获知自己的邻居节点的报文情况.

如果关键节点没有收到紧邻的下游关键节点传送来的 ACK 报文，说明在自己和紧邻的下游关键节点之间有恶意节点存在，在接收到下游的某个普通节点传送过来的 ACK 报文后，它将该报文通过备份路径传送给紧邻的下游关键节点，紧邻的下游关键节点又将该报文向其上游方向传送.

如果关键节点收到段中上游节点传送来的 NACK 报文，它将该报文通过备份路径传送给其紧邻的上游关键节点，紧邻的上游节点会将该报文向下游方向传送.

(3) 恶意节点

恶意节点能识别链路检测控制报文，它总是选择最有利于自己的策略来向其它节点谎报自己的转发报文情况.

图 2 给出了一个实例，其中 M 为恶意节点，在不同节点间按序号发送的报文情况说明了链路检测的过程.

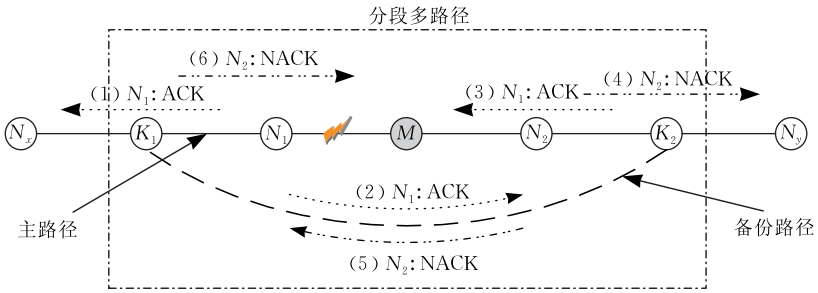


图 2 基本链路状态检测

4.5 邻居节点状态信息维护

每个节点维护一份邻居节点状态表，如下所示：

Neighbor ID	Total packets	Dropped packets	Type I malicious behaviors	Type II malicious behaviors
				Adjacent Node Number

记录各邻居节点转发报文数量以及由于 I、II 类恶意行为分别造成的报文丢弃数量，由于 II 类恶意行为通常是由于相邻节点所声称转发报文数不一致造成，还需记录与此节点相邻的节点 *ID*. 节点依据收到的 ACK 和 NACK 中的 *HopToSrc* 字段以及 NACK 报文中的 *DropNum* 字段，记录邻居节点的报文转发情况.

4.6 恶意行为的检测

4.6.1 单个恶意节点情况下的检测

本小节考虑当路径上存在单个恶意节点时，通信链路实时分析协议的检测结果. 假设 M 为恶意节点，分情况讨论如下.

(1) 如果 M 拒绝发送 ACK 报文，情况如图 3(a)所示，检测结果为：$N_2 \rightarrow M$ 和 $N_3 \rightarrow M$ 是 I 类恶意行为，而 $N_1 \rightarrow N_2$ 和 $N_4 \rightarrow N_3$ 为 II 类恶意行为.

(2) 如果 M 选择发送链路检测控制报文，如图 3(b)所示，可分为 3 种情况：

(i) M 向上游节点发送 ACK 报文，检测出 $N_3 \rightarrow M$ 为 I 类恶意行为，而 $N_2 \rightarrow M$ 和 $N_4 \rightarrow N_3$ 为 II 类恶意行为.

(ii) M 向下游节点发送 NACK 报文，检测出 $N_2 \rightarrow M$ 为 I 类恶意行为，$N_1 \rightarrow N_2$ 和 $N_3 \rightarrow M$ 为 II 类恶意行为.

(iii) M 同时向上游和下游节点发送 ACK 报文和 NACK 报文，检测出 $N_2 \rightarrow M$ 和 $N_3 \rightarrow M$ 为绝对恶意行为.

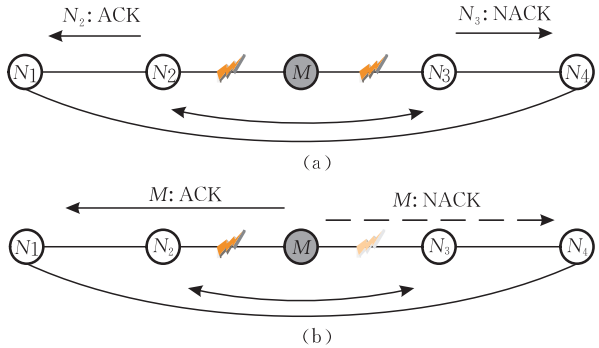


图 3 单个恶意节点场景下的链路状态检测

4.6.2　多个恶意节点情况下的检测

4.6.1 节中的原则和假设同样适用于检测链路上存在多个恶意节点的情况，但此时运行情况会更加复杂. 本小节通过分析不同场景，说明检测链路上存在多个恶意节点时的情况.

场景 1.　恶意节点位于多个不同段.

如图 4(a)所示，恶意节点 M_1 和 M_2 分别位于两个不同段中，如果 P_1 转发组中的所有报文一定时间后，没有收到目标节点发送的 ACK 报文，它自行生成 ACK 报文向上游节点传送. 对于 M_1 来说，选择发送 ACK 报文或者 NACK 报文是有利于躲避检测的策略(从分析角度来说，这是一种更强的攻击模型)，假设它向上游节点发送 ACK 报文，由于 P_1 收到了 M_1 发送的 ACK 报文，它向上转发 M_1 生成的 ACK 报文，该报文沿着 K_1-->K_2-->N_1→M_1 路径传递. N_1 收到该报文，发现 M_1 声称转发的报文个数与实际转发给自己的报文个数不一致，记录 M_1 的行为为 I 类恶意行为，并生成自己的 $NACK(n_1)$ 报文向自己的下游节点传递，其中 n_1 为节点 N_1 没有收到的该报文组内的报文个数.

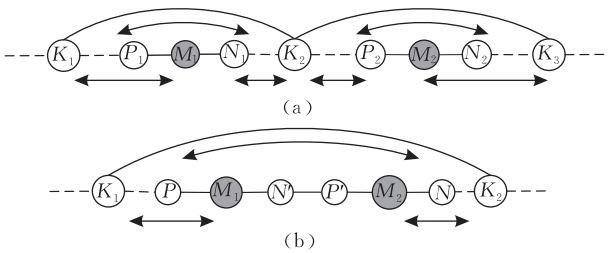


图 4　多个恶意节点场景下的链路状态检测

由 N_1 生成的 $NACK(n_1)$ 报文将沿着下面 3 条可能路径继续传递：

(1) N_1-->K_2-->K_1-->P_1→M_1. P_1 收到该报文，发现 M_1 与其下游邻居节点声称收到的报文个数不一致，记录 M_2 的行为为 II 类恶意行为. 同时沿途的其它节点收到该报文后，认为自己的邻居节点没有恶意的行为.

(2) N_1-->K_2-->P_2→M_2. M_2 收到该报文后，为了躲避检测，它可以选择自己生成 NACK 报文向下游节点传递，也可以选择仅仅不转发上游节点生成的 NACK 报文. 由于这两种决策对于恶意节点来说最终收益相同，为叙述方便，假定恶意节点仅仅不转发上游节点生成的 NACK 报文.

(3) N_1-->K_2-->K_3-->N_2→M_2. N_2 收到该报文后，发现自己没有收到的该报文组中的报文个数 $n_2>n_1$，由于它未收到 M_2 的生成的 NACK 报文，记录 M_2 的行为为 I 类恶意行为，并生成自己的 $NACK(n_2)$ 报文向自己的下游节点传递. 沿途的下游节点收到该报文后，认为自己的邻居节点没有恶意的行为.

当 K_3 收到 $NACK(n_2)$ 报文后，该报文沿着 K_3-->K_2-->P_2 路径向上游传递. 当 P_2 收到该报文后，发现 M_2 与其下游邻居节点声称收到的报文个数不一致，记录 M_2 的行为为 II 类恶意行为. 同时沿途的其它节点收到该报文后，认为自己的邻居节点没有恶意的行为.

场景 2.　段中有多个恶意节点.

如图 4(b)所示，假设恶意节点 M_1 为了躲避检测，首先向上游节点发送 ACK 报文，该报文将沿着 M_1→P-->K_1-->K_2-->N→M_2 路径传递. 当该 ACK 报文到达恶意节点 M_2 后，M_2 根据有利于自己的原则，向下游节点发送 $NACK(n_1)$ 报文.

M_2 生成的 $NACK(n_1)$ 一方面由关键节点 K_2 通过主路径和备份路径向下游继续传递，另一方面沿着 M_2→N-->K_2-->K_1-->P→M_1 路径传递. 此过程中，N 和 P 分别发现 M_2 和 M_1 与它们邻居节点声称的组报文转发个数不一致，记录它们的行为为 II 类恶意行为，同时认为在路径 $M_1\cdots M_2$ 上有恶意节点.

当关键节点只能定位到一条路径上存在恶意节点时，例如 $M_1\cdots M_2$，将随机选择除这条路径两个端点外的任一节点 N_i，发起到该节点的路由请求，在路由请求中明确声明路径需要规避的节点 M_i，通过转发该路径关键节点收到的 ACK 和 NACK 报文给 N_i，N_i 将该报文向上下游节点双向传递，最终 N'、P' 认为 N'→M_1 和 P'→M_2 为 I 类恶意行为.

5　分布式节点类型判定算法

5.1　概　述

通过通信链路状态实时分析协议，节点获得邻居节点的报文转发状态信息，节点根据这些信息分析节点行为是否异常. 但是该过程只是单个节点的个体行为，对于恶意节点的最终认定还需要考虑到网络中其它节点对该节点行为的评判结论，为此本节设计了分布式的节点类型判定算法，该算法建立节点类型判定的数学模型，通过分析和处理邻居节点丢包率，依据大多数选举原则来判定节点是否为恶意节点.

在整个算法中，节点可发送和接收“通告邻居节

点丢包率”报文、“申述”报文和“通告节点状态”报文，用于通告和获取相关信息.

5.2 邻居节点丢包率及其通告

根据邻居节点状态表，假设节点共转发了 n 个报文，在其邻居节点所在的链路上共有 m 个报文被丢弃，这 m 个丢弃的报文归结为由邻居节点两种类型的恶意行为造成，其中 I 类恶意行为造成了 m_1 个报文的丢弃，II 类恶意行为造成了 m_2 个报文的丢弃，$m=m_1+m_2$. 节点可进一步计算各邻居节点的加权报文丢弃数：$m'=\begin{cases} m_1\times\alpha+\beta\times m_2, & m'<n \\ n, & m'>n \end{cases}$，其中，$\alpha$ 的值根据实际的情况加以选取，通常 $\alpha>1$，$\beta\leq 1$.

由于 I 类恶意行为很可能是由于恶意节点故意丢包造成，需放大 I 类恶意行为丢包程度. 报文丢弃率重新计算为 $P_d=m'/n$，因此每个节点获得其所有 k 个邻居节点的报文丢弃率 $p_d^1,p_d^2,\cdots,p_d^i,\cdots,p_d^k$.

节点据此生成“通告邻居节点丢包率”报文，报文格式如下所示：

Number	Neighbor ID_1	Drop rate	…	Neighbor ID_n	Drop rate

节点定期发送该报文给自己两跳范围内的邻居节点. 在此过程中，恶意节点可能会不转发该报文，但这是对其不利的策略，如图 5 所示，即使恶意节点 C 不泛洪 A 的通告报文，B 也可通过它和 A 的共同邻居节点 D 获得 A 的通告报文. 其次如果 B 总是收不到 C 转发的其它邻居节点的通告报文，而其它邻居节点通告报文中又包含 C 的丢包记录，那么 C 的行为就十分可疑. 为了避免出现这种情况，C 选择转发邻居节点的通告报文是明智的选择.

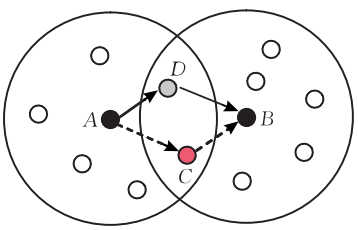


图 5 邻居节点丢包率通告示意图

当节点从自己邻居节点收到“通告邻居节点丢包率”报文后，记录邻居节点反馈的关于自己的报文丢包率 $\tilde{p}_d^1,\tilde{p}_d^2,\cdots,\tilde{p}_d^i,\cdots,\tilde{p}_d^k$，同时为每个邻居节点建立邻居节点丢包率表.

5.3 节点类型判定及恶意节点隔离

5.3.1 丢包率的调整

节点计算邻居节点反馈的关于自己的报文丢包率 $\tilde{p}_d^i$ 和本身记录的邻居节点丢包率 p_d^i 之间的差值 $Diff^i=|\tilde{p}_d^i-p_d^i|$，检测该偏差程度是否异常. 由于节点的邻居节点个数通常有限，选择肖维勒准则来检测. 首先计算差值样本的均值和方差：

$$\mu_{Diff}=\frac{1}{k}\sum_{i=1}^{k}Diff^i,\quad S_{Diff}=\sqrt{\frac{\sum_{i=1}^{k}(Diff_i-\mu_{Diff})^2}{k-1}}.$$

如果 $Diff_i>(\mu_{Diff}+G_k\cdot S_{Diff})$，则可判断其为异常值，其中 G_k 可根据实际的邻居节点个数来选择，比较典型有 $G_6=1.73$，$G_8=1.86$，$G_{10}=1.96$，$G_{12}=2.03$. 根据判定的宽严程度，G_k 还可以适当的加以调整. 若 $Diff^i$ 异常，则节点发送“申述”报文泛洪给 2 跳范围内的节点，并修改该邻居节点的丢包率 p_d^i 为其反馈的自己的丢包率 $\tilde{p}_d^i$，若节点被一半以上的邻居节点申述，则该节点将会被列入恶意节点名单. 该策略可有效地防止恶意节点发送虚假的邻居节点丢包率值，否则将面临提高自己丢包率值和被申述的风险.

5.3.2 根据丢包率检测异常节点

依据邻居节点丢包率表，可根据 $\mu_j=\frac{1}{n}\sum_{i=1}^{n}p_d^i$，$\tilde{\mu}=\frac{1}{k}\sum_{i=1}^{k}\mu_j$ 和 $S=\sqrt{\frac{\sum_{i=1}^{k}(\mu_j-\tilde{\mu})^2}{k-1}}$，计算出 k 个邻居节点的平均丢包率 $\mu_C,\mu_D,\cdots,\mu_j,\cdots,\mu_k$，以及总平均丢包率 $\tilde{\mu}$ 和方差 S.

同样依据肖维勒准则，依次对 μ_j 做异常值分析，如果 $\mu_j>(\tilde{\mu}+G_k\cdot S)$，则可判断其为异常值，根据判定的宽严程度，$G_k$ 可以适当地加以调整.

5.3.3 节点类型判定

根据节点平均报文丢弃率 μ_j 和报文转发数来判定节点是否为可信节点. 若节点的平均丢包率 μ_j 为异常值，将认定该节点为报文转发过程中的异常节点. 若节点的报文转发数高于设定的阈值且报文丢弃率低于设定的阈值，判定该节点为可信候选节点，阈值可以根据网络的类型在部署时预先设定.

根据统计分析出来的异常节点和可信候选节点列表，节点生成“通告节点状态”报文，在 3 跳范围内泛洪. 对于任意一个节点 N_i，若有 k 个邻居节点，且有 $k/2$ 以上个节点认为该节点为异常节点，则认定该节点为恶意节点；如果有 $k/2$ 以上个节点认为该节点为可信候选节点，则认定该节点为可信节点.

5.3.4 恶意节点的隔离

当一个节点被选举为恶意节点后，其邻居节点不再转发该节点的报文，将该节点从网络中隔离出

去，同时其 2 跳邻居节点检查邻居节点状态表，消除邻居节点中由于该恶意节点导致的 II 类恶意行为记录，重新计算所有邻居节点的丢包率.

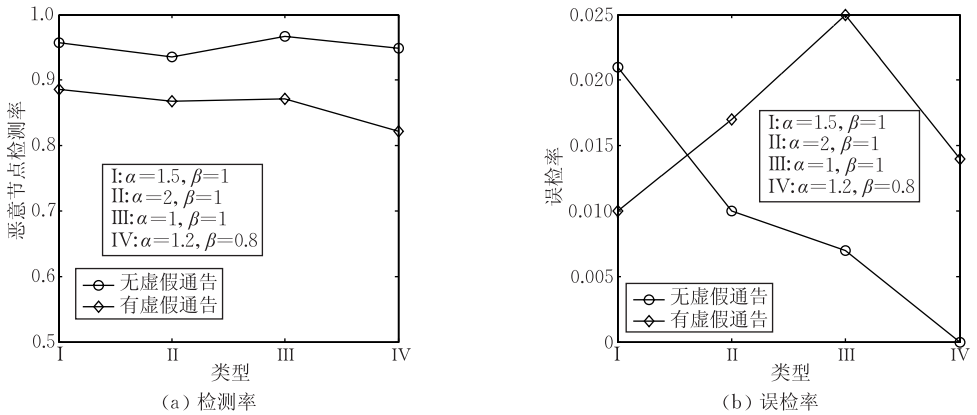
6　仿真实验和分析

仿真实验在 2000×2000 范围下，构造出不同节点度下的测试场景，节点度指该节点 1 跳范围内邻居节点个数. 不同总数的节点按照均匀分布部署，按不同比例选择部分节点充当恶意节点随机均匀，每轮随机选择生成网络中总节点数一半的通信节点对相互发送 50 个报文，网络中正常节点的报文丢弃率 $0\leq P_d\leq 0.1$，恶意节点的报文丢弃率 $0\leq P_d\leq 0.5$. 本节仿真恶意节点不发送虚假的邻居节点丢包率信息以及恶意节点选择邻居节点中的 20% 发送虚假的[0.5,1]之间邻居节点丢包率信息两种情况.

评价算法性能的主要指标是恶意节点的检测率和恶意节点的误检率. 检测率是指被检测出来的恶意节点个数占网络中实际的全部恶意节点个数的比例. 误检率是指正常节点被误检为恶意节点的个数占整个网络中正常节点的比例.

6.1　α 和 β 对检测性能的影响

α,β 分别为 I 类恶意行为和 II 类恶意行为在计算加权的报文丢弃率时的权重，采用加权的方法计算报文丢弃率主要是为了加大对 I 类恶意行为的惩罚，因为 I 类恶意行为通常是由于恶意节点故意丢弃报文造成的，虽然它可能由于网络报文传输中的正常错误引起，例如内存拷贝错误和协议栈的报文缓冲区溢出，但是相对于恶意节点的故意丢包行为来说，正常错误产生的概率要低得多. α/β 值越大说明对 I 类恶意行为的惩罚相对于 II 类恶意行为来说力度越大. 为了选择合适的 α,β 值，在节点度 $d=15$，恶意节点比例为 0.2 的情况下，分别选择 4 对不同的(α,β)值进行实验，其中 I:($\alpha=1.5,\beta=1$)；II:($\alpha=2,\beta=1$)；III:($\alpha=1,\beta=1$)；IV:($\alpha=1.2,\beta=0.8$)，实验结果如图 6 所示.



(a) 检测率　　(b) 误检率

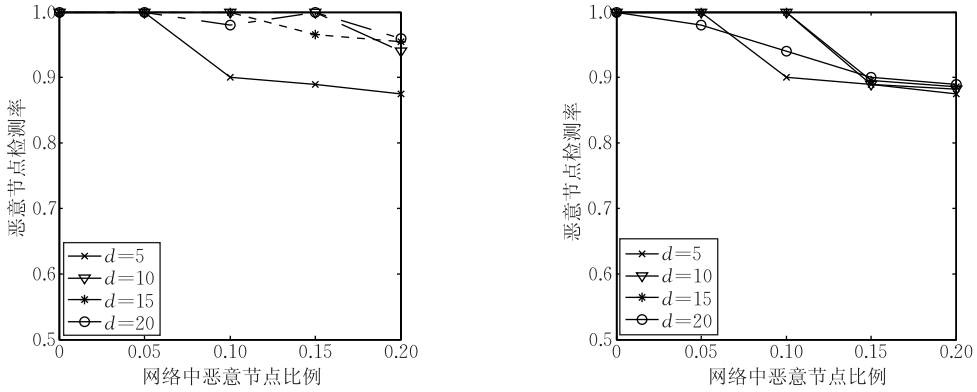
图 6　不同 α,β 值下检测率和误检率

图 6 表示了不同 α,β 值下的恶意节点检测率和误检率，可以看出在 I:($\alpha=1.5,\beta=1$)条件下，算法在检测率和误检率上综合性能表现最好，尤其是在恶意节点发送虚假的邻居节点丢包率信息时. 在 $\beta=1$的情况下，α 取值过大会相对削弱 II 类恶意行为所表现产生的异常度，α 取值过小则没有有力地惩罚恶意节点常犯的 I 类恶意行为，相比较而言，$\alpha=1.5$算法性能最好，因此本节后面的实验均在($\alpha=1.5,\beta=1$)条件下进行.

6.2　不同恶意节点比率下的检测率

无论恶意节点是否发送虚假的邻居节点丢包率信息，算法的检测率均受到节点度和恶意节点比例两方面因素的影响，图 7 表明节点度越大，恶意节点的检测率越高，因为节点度越大，邻居节点报文转发率信息数据越丰富，方便了统计分析；恶意节点的检测率随着恶意节点在网络中所占比例的增加有所下降，因为恶意节点的增加导致了网络中节点报文转发率的异常数据增多，干扰了统计分析过程. 恶意节点发送虚假的邻居节点报文转发率信息对检测率也造成了一定的影响，尤其在恶意节点比例较高的情况下，检测率降低了 5% 左右.

总之，本文算法的恶意节点检测率性能良好，尤其在恶意节点所占比例较低的情况下，接近 100%. 随着恶意节点所占比例的提高和恶意节点发送虚假的邻居节点报文转发率信息，恶意节点的检测率有所下降，但仍保持在 90% 左右.



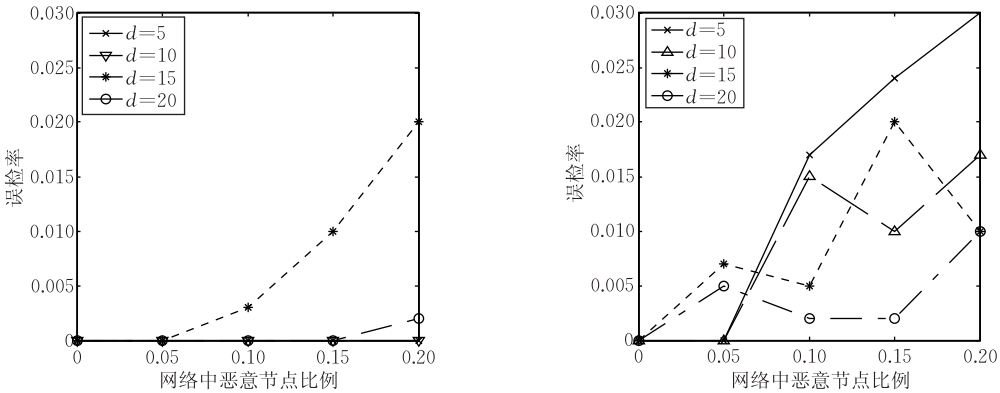
(a) 没有虚假通告报文 (b) 有虚假通告报文

图 7 不同恶意节点比例下的检测率

6.3 不同恶意节点比率下的误检率

同恶意节点检测率一样,本文算法的误检率也受到节点度和恶意节点比例的影响.从整体上来看,节点度越大,恶意节点所占比例越低,误检率越低.节点度越大,样本数据集就越大,就越能准确地通过统计分析手段来检测出恶意节点,减少误判率.

图 8 表明本文算法误检率较低,即使有恶意节点发送虚假的"通告邻居节点丢包率"报文,误检率最高为 3%.当节点平均度达到 20 时,即使恶意节点比率达到 20%,误检率约为 1%.图 8 还表明本文算法误检率受恶意节点所占比率影响不大,因为该算法依靠所有邻居节点综合信息的反馈来判定恶意节点,削弱了由恶意节点产生的干扰信息的破坏能力.



(a) 没有虚假通告报文 (b) 有虚假通告报文

图 8 不同恶意节点比例下的误检率

7 结束语

本文分析了无线自组织网络环境下来自于网络内部节点的丢弃报文攻击的危害性.通过分段多路径协议构成的分段上的主路径和备份路径,利用网络层报文转发的成组 ACK 机制,本文设计了一种新的链路状态检测协议,分析了链路上存在单个恶意节点和多个恶意节点场景下,获取邻居节点报文状态的过程.本文对不同类型的报文丢弃行为分类,计算出节点加权报文丢弃率,在获得综合的邻居节点报文转发状态的情况下,根据肖维勒准则判定报文转发过程中的异常节点,通过局部选举方法来判定最终的恶意节点和可信节点.本文还在不同节点度和恶意节点比例的网络场景下,随机生成网络背景流量,测试本文算法在恶意节点检测率和误判率上的性能表现,实验结果表现本算法性能良好.在今后的工作中,我们还将进一步优化异常节点判定算法,并考虑节点移动性对本文算法产生的影响.

参 考 文 献

[1] Shao M, Zhu S. A cross-layer dropping attack in video streaming over Ad Hoc networks//Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (Securecomm'08). Istanbul, Turkey, 2008: 1-8(Article No.: 25)

[2] Marti S, Giuli T, Lai K et al. Mitigating routing misbehavior in mobile Ad Hoc networks//Proceedings of the 6th International Conference on Mobile Computing and Networking (Mobicom'00). Boston, USA, 2000: 255-265

[3] Lee S, Choi Y. A resilient packet-forwarding scheme against maliciously packet-dropping nodes in sensor networks//Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06). Alexandria, USA, 2006: 59-70

[4] Khalil I, Bagchi S. Mispar: Mitigating stealthy packet dropping in local-monitored multi-hop wireless Ad Hoc networks//Proceedings of the 4th International Conference on Security and Privacy in Communication Networks (Securecomm'08). Istanbul, Turkey, 2008: 1-10(Article No.: 28)

[5] Liu K. An acknowledgment-based approach for the detection of routing misbehavior in MANETs. IEEE Transactions on Mobile Computing, 2007, 6(5): 536-550

[6] Kozma W, Lazos L. REAct: Resource-efficient accountability for node misbehavior in Ad Hoc networks based on random audits//Proceedings of the 2nd ACM Conference on Wireless network security (Wisec'09). Zurich, Switzerland, 2009: 103-110

[7] Awerbuch B, Curtmola R, Nita-Rotaru C. ODSBR: An on-demand secure Byzantine resilient routing protocol for wireless Ad Hoc networks. ACM Transactions on Information Systems Security, 2008, 10(4): 1-35

[8] Zhang X, Jain A, Perrig A. Packet-dropping adversary identification for data plane security//Proceedings of the 4th International Conference on Emerging Networking Experiments and Technologies (Conext'08). Madrid, Spain, 2008: 1-12 (Article No.: 24)

[9] Zouridaki C, Mark B, Hejmo M et al. A quantitative trust establishment framework for reliable data packet delivery in MANETs//Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (Sasn'05). Alexandria, USA, 2005: 1-10

[10] Zouridaki C, Mark B, Hejmo M et al. Robust cooperative trust establishment for MANETs//Proceedings of the 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (Sasn'06). Alexandria, USA, 2006: 23-34

[11] Sun Y, Han Z, Yu W et al. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks//Proceedings of the 25th IEEE International Conference on Computer Communications (Infocom'06). Barcelona, Spain, 2006: 1-13

[12] Sun Y, Yang Y. Trust establishment in distributed networks: Analysis and modeling//Proceedings of the IEEE International Conference on Communications (Icc'07). Glasgow, Scotland, 2007: 1266-1273

[13] Liu F, Cheng X, Chen D. Insider attacker detection in wireless sensor networks//Proceedings of the 26th IEEE International Conference on Computer Communications(Infocom'07). Anchorage, USA, 2007: 1937-1945

[14] Brown J, Du X. Detection of selective forwarding attacks in heterogeneous sensor networks//Proceedings of the IEEE International Conference on Communications (ICC'08). Beijing, China, 2008: 1583-1587

[15] Wang C, Feng T, Kim J et al. Catching packet droppers and modifiers in wireless sensor networks//Proceedings of 6th IEEE Conference on Sensor, Mesh & Ad Hoc Communication Networks (Secon'09). Rome, Italy, 2009: 529-537

[16] Wang B, Wei W, Zeng W et al. Fault localization using passive end-to-end measurement and sequential testing for wireless sensor networks//Proceedings of the 6th IEEE Conference on Sensor, Mesh & Ad Hoc Communication Networks (Secon'09). Rome, Italy, 2009: 225-234

ZHANG Zhong-Ke, born in 1980, Ph. D. candidate. His research interests include security, trust model and privacy protection issues in wireless ad hoc and sensor networks.

Wang Yun, born in 1967, professor, Ph. D. supervisor. Her research interests include distributed computing, fault-tolerance and wireless ad hoc and sensor networks.

Background

This paper focuses on the research of defending against packet dropping attack from inside attacker in wireless ad hoc networks. Wireless ad hoc networks are vulnerable to a wide range of security attacks owing to the open nature of wireless channels, the characters of self-organization and the uncertainty of node behavior. Moreover, different from wired networks, there are no specific routers in wireless ad hoc networks and any node in such networks can act as both router and terminal. This makes malicious nodes easy to present themselves in routes between source-destination pairs, and then drop data packets arbitrarily. Such a kind of attack, called insider packet dropping attack, will dramatically degrade network performance. However, neither transport layer nor network layer protocols of existing network protocol stacks are able to defend against it efficiently. Therefore, how to defend against insider packet dropping attack in the

network layer becomes a hot research topic in recent years. However, some research results are confined to certain routing protocols, or some cannot deal with attack under strong adversary model, or some have limitations of inefficiency and low detection rate. This paper designed a new secure communication model to effectively defend against insider packet dropping attack, including Real-time Link Status Analysis protocol (RLSA) and Distributed Node Classification algorithm (DNC). RLSA obtained packet forwarding status of neighbors in real time. DNC identified abnormal behavior of nodes based on RLSA and theoretical analysis model, and then classified nodes into normal and malicious types. Compared to existing works, the approach in this paper is a more general method, which could be built on the top of any kind of routing protocols. It is also characteristic of simplicity yet effectiveness, as well as real-timing analysis and detection with a high detection rate and a low false alarm rate, and can be easily applied into wireless ad hoc networks.

This research work is supported by the Natural Science Foundation of China under grant No. 60973122 and the National Basic Research Program of China (973 Program) under grant No. 2009CB320705. These grants focus on wireless ad hoc and sensor networks and pervasive computing, and aim to enhance network reliability, security, improve network service quality and confidence assurance and so on. The research group has been working on wireless networks for years, and some research results have been published in world-wide journals and conferences, including InfoCom, ICDCS, LCN, Science in China, IJAHUC, Cluster Computing and Chinese Journal of Computer and so on. This paper addresses the issue of improving wireless network security and reliability.