文章编号:1001 - 9081(2007)10 - 2464 - 06

战术互联网同质层基于信任评估的安全分簇算法

张 晗[1],万明杰[1],王寒凝[2]
(1. 解放军防空兵指挥学院 防空导弹系,郑州 450052; 2. 北京交通大学 交通运输学院,北京 100044)
(zh_swallow@sina.com)

摘 要:提出一种适用于构建战术互联网同质网层分级结构的安全分簇算法(SCABTE),给出簇形成和簇维护的详细设计策略。最后从入侵节点的角度分析其安全性,并用 NS-2 构建了战术互联网仿真环境在节点传输距离、节点移动速度、网络规模变化条件下验证算法的可用性。

关键词:战术互联网;网络安全;信任评估;分簇算法
中图分类号: TP301.6 **文献标志码**:A

Secure clustering algorithm based on trust evaluation in tactical Internet homogeneous layer

ZHANG Han[1], WAN Ming-jie[1], WANG Han-ning[2]
(1. *Department of Air Defence Missible, College of Air Defence Command College, Zhengzhou Henan* 450052, *China*;
2. *School of Traffic and Transportation, Beijing Jiaotong University, Beijing* 100044, *China*)

Abstract: A kind of secure clustering algorithm Secure Clustering Algorithm Based on Trust Evaluation (SCABTE) was proposed based on trust evaluation. It was suitable in constructing the tactical Internet homogeneous layer network structure. The detailed design tactics of cluster forming and maintenance were provided. The security problem was analyzed from the point of view of invasion node. Finally, the usability of the algorithm was validated in the NS-2 simulation environment adopting Reference Point Group Mobility Model (RPGMM).

Key words: tactical Internet; security network; trust evaluation; clustering algorithm

0 引言

战术互联网中从通信设备构成的角度划分为异质网络和同质网络[1]。异质网络所含通信设备呈多元化,可包括各类频率功率不同的数字通信电台[2];而同质网络一般由发送频率功率一致的同类电台设备构成。同质网层节点(例如超短波电台)通常由簇首和网关节点构成虚拟骨干网[3],此时需要有较为复杂的簇首选举和簇维护机制形成分簇算法。

在战术互联网中,簇结构的安全性将直接影响到其上应用,如路由、安全性。参与破坏簇结构的节点根据其来源可以分为恶意节点和合谋节点两类:试图破坏网络的未经授权的节点称为恶意节点,敌对方可以通过部署恶意节点实施攻击,如监听、消息重放、篡改以及伪造簇生成消息等;将经过授权但被敌对方接管的节点称为合谋节点。现有簇生成算法对安全性考虑不足,与战术互联网中的路由安全问题类似,目前的簇生成算法均假设应用环境为可信网络。然而,在簇生成过程中,一个恶意的节点能够通过发布虚假的信息(如节点的连接度、链路稳定性等),从而导致簇结构的稳定性差、负载不均衡等多种问题。如果一个恶意节点被选为网关,那么所有经过该网关的报文都有可能被窃听,或者被其故意丢弃,导致报文的重发、整个网络的拥塞等。

由于合谋节点能够正常完成所有授权节点的操作,因此检测合谋节点十分困难,尤其在合谋节点不参与恶意攻击的情况下。本文仅考虑参与恶意破坏的合谋节点,同时算法能够将战术互联网入侵检测系统发现的合谋节点从网络中剔除。

针对现有簇生成算法对簇生成安全性考虑不足的问题,本文在分析敌对环境下簇生成算法安全需求的基础上,针对军事通信网络信任关系易建立不易保持的特点,提出基于信任评估的安全分簇算法(Secure Clustering Algorithm Based on Trust Evaluation,SCABTE),并给出该算法所涉及的信任评估公式以及簇的形成与维护的具体策略。SCABTE 算法为单跳主动簇生成算法,能够有效防止不良节点对簇结构的破坏,从而减少不良节点对整个网络的安全威胁。

1 算法设计

1.1 网络环境

本文对所研究的网络环境做如下假设:

1)网络中所有节点独立且对等,网络采用全分布操作;

2)网络在大部分时间内连通,网络分割只是偶尔发生的事件;

3)节点采用无向天线,且天线传输能力相同;

4)每个节点具有唯一的标识,且具有单跳邻居节点发现机制;

5)节点可在网络中自由移动;

6)网络采用战术互联网络路由协议支持节点间的多跳通信。

所有链路均为双向链路且链路层能够检测出随机损坏的报文,并且节点的时钟同步。此外物理层的安全攻击以及介

收稿日期:2007 - 05 - 09;**修回日期**:2007 - 06 - 28。 **基金项目**:国防预研项目。

作者简介:张晗(1980 -),女,河南郑州人,硕士,主要研究方向:信息安全、电子商务; 万明杰(1966 -),男,河南郑州人,副教授,博士,主要研究方向:运筹学、测控; 王寒凝(1978 -),女,安徽肖县人,博士研究生,主要研究方向:数据挖掘、信息安全。

质访问控制层的拒绝服务攻击不在本文的讨论范围内。

1.2　算法思想

算法基于战术互联网网络初始化时依据由节点的隶属关系确定的初始信任值构建初始分级结构。战术互联网具备网络功能开始执行任务以后,节点可以根据自己的经验值评估其邻居节点的信任值。当存在一个簇的簇内成员对其簇首的信任度产生置疑时可以发送申告该簇簇首的广播消息。将申告的次数作为引发簇首竞争机制的门限值,该值可根据簇的具体情况设定(例如簇成员的数目,执行任务的安全级别)。此时簇内成员在计算其邻居节点的信任值完成后向信任值最高的节点发送推荐证书,持有最多推荐证书的节点作为簇首,参选的节点作为簇成员,再次构成分级结构。簇内节点发现簇内其他不良行为节点同样可以发送通告广播消息。簇首持有最高的信任值同时负责给簇内成员发放信任值证书。算法还具备簇维护功能,包括对簇首和成员节点的维护。同时算法在簇生成消息中引入了证据信息,使算法不但能够避免恶意节点的破坏,而且能够防止单个合谋节点通过恶意的行为破坏簇结构的生成和维护过程。除此之外,由于链路的产生、故障、删除、恢复对分簇结构造成的影响包含在节点的出现、消失和移动造成的结果中,因此不再单独考虑。算法功能结构如图 1 所示。

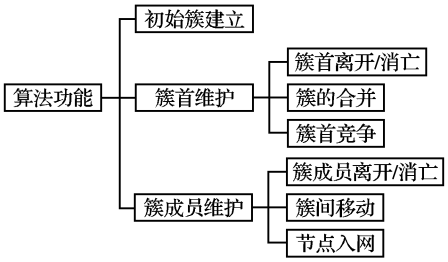

图 1　SCABTE 算法功能结构

1.3　SCABTE 算法中的密钥

算法假设战术互联网的初始化阶段是在安全、可控的环境下进行的,且初始化阶段所有的节点都是可信的。此时可信第三方作为离线的可信管理中心仅仅在网络初始化阶段引入。

1.3.1　主密钥

算法采用文献[4]提出的移动自组织网络证书服务机制实现簇生成消息的鉴别和授权节点的认证以及保证消息的完整性。该机制在网络初始化阶段由离线的可信第三方基于 RSA 算法为网络随机产生主密钥,同时为每个节点生成基于 RSA 算法的公/私钥对 $\{sk_i, pk_i\}$,用于节点间的认证和安全的交换会话密钥。

1.3.2　通信密钥

为了降低簇生成消息的加解密开销,可以将广播通信视为组通信的特例,使网络中所有节点共享同一个通信密钥。算法采用文献[5]中密钥管理为每个授权节点生成相同的通信密钥,以保证簇生成消息的私密性,并且通信密钥能够随着授权节点的变化及时更新。其思想是首先在初始化阶段由可信第三方为网络选择通信密钥种子函数 $g(x)$,而后结合通信种子密钥序号 *seed* 依据其定义的 *TEK* 算法生成通信密钥。

可信第三方同时为每个节点产生全网唯一的 ID 标识和初始信任值。由于它是唯一完整掌握系统私钥的机构,因此不能参与到系统运行中去,在系统初始化完成后立即退出系统。

1.4　SCABTE 算法要素

SCABTE 算法满足以下通用假设:

1)存在理想的信道接入协议,一个节点发出的消息能够被其所有的邻居节点在较短的时间内正确收到;

2)每个节点具有唯一的 ID,在网络初始化时,每个节点可以通过交互控制消息知道其邻居节点的初始信任值和 ID;

3)在分簇算法执行期间(簇结构建立时),网络的拓扑不发生改变。

战术互联网用图 $G = (V, E)$ 表示,其中 V 表示网络节点集合,且 $|V| = n$; E 表示网络通信链路的集合。当节点 v, u 在对方通信范围内时,用 $\{v, u\} \in E$ 表示节点间的通信链路。

1.4.1　符号与定义

节点分为未决状态、簇成员状态、未认证状态、等待状态、自荐状态、簇首状态和离开状态。

ID_i 为节点 i 的标识;$UnitID_i$ 为节点 i 隶属单位标识;$RankID_i$ 为节点 i 行政级别标识;$TrustValue_i$ 为节点 i 的初始值信任赋值;$TrustValue(i, j)$ 为节点 i 对节点 j 的信任评估值,由信任评估计算公式计算;$TrustValue\_threshold$ 为信任评估门限值;U 处于未决定状态的节点集合;$N(i)$ 为节点 i 的一跳邻居节点集合;$M(CHeaderID_i, state)$ 为簇首 i 的簇成员节点集合,初始值为 $\varnothing$,$State(i)$ 为状态位,为“1”表示簇首正常工作,为“0”表示簇首被合并时竞争失败/弹劾/离开/消亡,此时集合表示原稳定簇成员;$num(CHeaderID_i)$ 为簇成员数目;*chn* 为收到簇首响应信息数目,初始值为 nil;*comn* 为申告消息计数器,初始值为 nil,门限为 n(根据具体情况设定);$R\_certNum_i$ 为节点 i 持有的推荐证书数量;*preID* 为节点上一所属簇簇首 ID;*Round* 表示节点广播“Hello”探测消息时的跳数,初始值为“1”,节点发现邻居过程中,采用局部广播发现邻居节点,用于控制通信的范围;$CHeaderID_i$ 为节点 i 为簇首时的标识;m 为两簇相邻时引发簇首竞争的簇成员数量门限值;$State(i)$ 为节点 i 的状态;CH_i 为节点 i 所属簇首的 ID,初始值为 nil;$Weight_i$ 为节点 i 的权值,为信任值、节点标识、节点行政级别标识的三元组 $\{R\_certNum_i, ID_i, RankID_i\}$;若节点 u、v 的权值满足以下条件:

$$(R\_certNum_v > R\_certNum_u) \vee ((R\_certNum_v = R\_certNum_u) \wedge (RankID_v > RankID_u)) \vee (R\_certNum_v = R\_certNum_u) \wedge (RankID_v > RankID_u) \wedge (ID_v < ID_u))$$

则称节点 u 的权值大于 v 的权值,用 $w_i > w_j$ 表示;$sk_i(hash)$ 为利用私钥 sk_i 对消息的摘要信息 *Hash* 签名。

1.4.2　消息类型

HELLO　处于未决定状态的节点间定时交换的存在消息,其格式为:

$$HELLO_i = \left(ID_i \mid UnitID_i \mid HELLO_j \mid RankID_i \mid hello \mid TrustValue_i \mid curtime \mid sk_i(hash)\right)$$

其中若 $HELLO_j$ 不为空,则满足 $w_j > w_i$ 或 $TrustValue_j > TrustValue_i$;

$JOIN(v, u)$　节点 u 成为节点 v 为簇首的簇的成员节点时,周期广播此消息,格式为:

$$JOIN(v, u) = \left(ID_u \mid curtime \mid TrustValue_u \mid ID_v \mid member \mid sk_i(hash)\right)$$

$CH(v)$　节点 v 为簇首时发送的定时局部广播消息,其格式为:

$$CH(v) = \left(ID_v \mid UnitID_v \mid curtime \mid TrustValue_v \mid clusterhead \mid memberInfo \mid EvidenceMessage \mid sk_v(hash)\right)$$

此时 *EvidenceMessage* 为簇首收到的推荐证书；

AuthenticateMessage(v, u) 簇首 v 给节点 u 发送的认证消息，其格式如下：

$$AuthenticateMessage(v, u) = \left(ID_v \mid ID_u \mid curtime \mid T\_Certificate \mid R\_Certificate \mid sk_v(hash)\right)$$

NOTIFY(v, u) 当节点 u 发现节点 v 为合谋节点时，向整个网络广播此消息，当节点 v 为簇首时触发簇内簇首竞争，消息格式如下：

$$NOTIFY(u, v) = \left(ID_v \mid ID_u \mid EvidenceMessage \mid curtime \mid sk_u(hash)\right)$$

此时 *EvidenceMessage* 为可以证明 u 为合谋节点的依据信息。

JoinMessage(v, u) 当节点 u 收到 $CH(v)$ 消息后发送的加入消息，其格式为：

$$JoinMessage(v, u) = \left(ID_u \mid UnitID_u \mid RankID_u \mid ID_v \mid preID \mid TrustValue_i \mid T\_Certificate \mid R\_Certificate \mid curtime \mid sk_u(hash)\right)$$

在节点第一次入网则其中 *preID*，*T_Certificate* 可以为空；

ComplainMessage(v, u) 当节点 u 对其所在簇簇首的可信度有置疑时发送的申告消息，为局部广播消息，其格式为：

$$ComplainMessage(v, u) = \left(ID_u \mid ID_v \mid TrustValue(u, v) \mid complain \mid sk_u(hash)\right)$$

RecomMessage(v, u) 为节点 v 对其邻居节点 u 的推荐消息，其格式如下：

$$RecomMessage(v, u) = \left(ID_u \mid ID_v \mid TrustValue(u, v) \mid recommend \mid R\_Certificate \mid sk_v(hash)\right)$$

$LeaveCommand_v$ 命令节点 v 离开的指令，其格式如下：

$$LeaveCommand_v = \left(ID_v \mid leave \mid targetUnitID\right)$$

其中 *targetUnitID* 可以为空；

$LeaveMessage_v$ 节点离开原簇时向簇首发送的离开消息，格式如下：

$$LeaveMessage_v = \left(ID_v \mid leave \mid sk_v(hash)\right)$$

1.4.3 证书

Recommendation Certificate 推荐证书，表示节点 v 对节点 u 的推荐，其格式为：

$$R\_Certificate(v, u) = \left(ID_v \mid ID_u \mid Recommend \mid ID_v'PUB\_key \mid createTime \mid Validation \mid sk_v(hash)\right)$$

TrustValue Certificate 信任证书，由簇首 v 发放给簇成员 u，其格式为：

$$T\_Certificate(v, u) = \left(CHeaderID_v \mid ID_u \mid TrustValue(v,$$

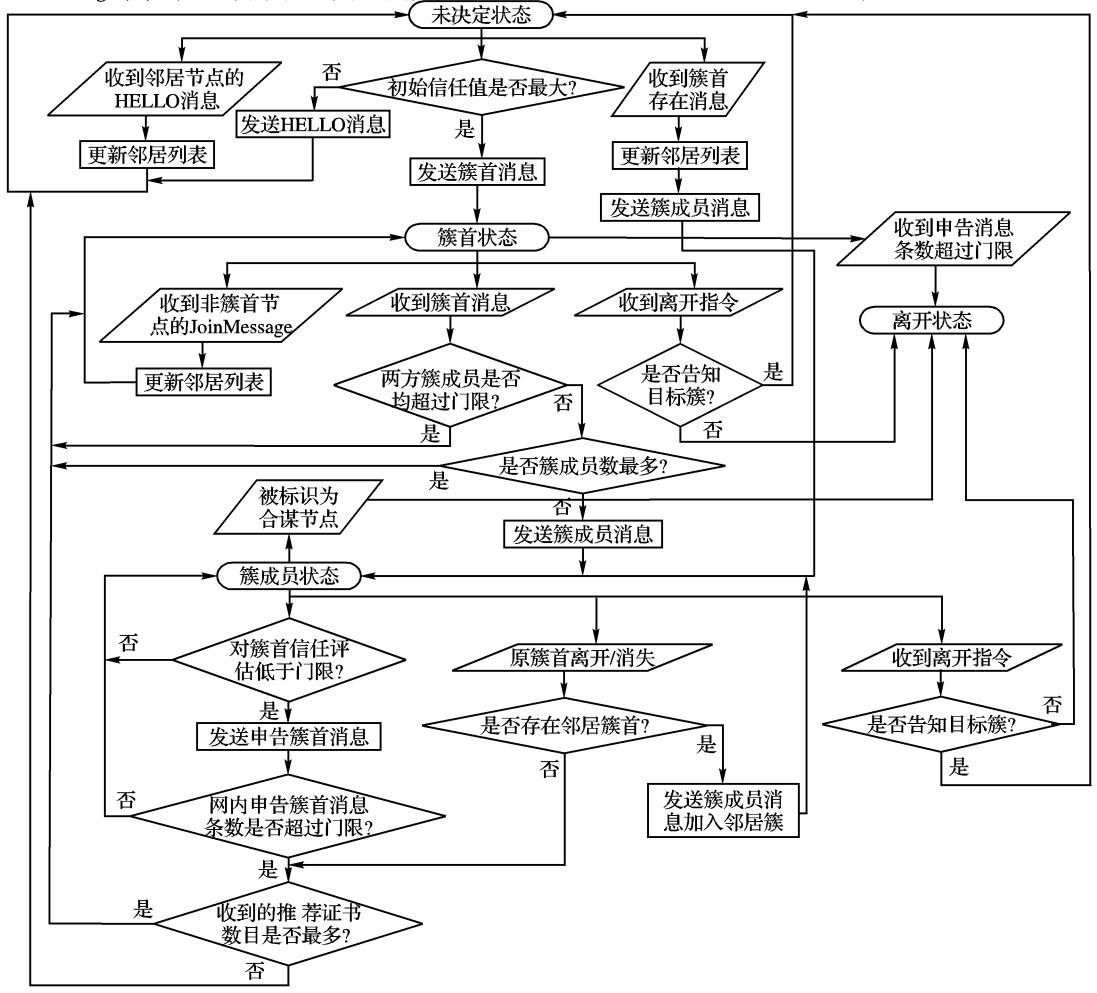

图 2 节点在簇形成和簇维护过程中的状态转换流程

$u) \mid CHeaderID' PUB\_key \mid createTime \mid Validation \mid sk_v(hash)\big)$

1.4.4　定时器

T_e 为等待状态定时器。用于避免节点同时发送局部广播而导致无线信道过度拥塞以及报文丢失。

T_n 为等待邻居组成员节点响应定时器。当簇成员节点发送邻居发现消息后,等待邻居成员节点响应。该定时器的等待时间随 *Round* 的增加而逐渐增加。

T_h 表示广播周期,且 $T_e > T_h$。

若节点在等待时间内未能从邻居节点接收到消息,表明链路断开。

1.5　算法描述

算法由消息驱动即节点根据其接收到的消息执行特定的动作。节点状态转换如图 2 所示。

1.5.1　*初始簇建立*

在初始状态下,集合 $N(i)$, $M(CHeaderID_i, state)$ 均为 $\varnothing$, $i = 1, 2, \cdots, n$,节点利用定时器 T_h 周期广播 *HELLO* 消息。节点接收到周期广播消息后,根据消息的签名验证消息的来源以及完整性,而后创建邻居节点集合 $N(i)$,当定时器 T_e 超时后,启动簇生成算法,比较初始信任值 *TrustValue*。未决节点将自己的信任值和一跳未决邻居节点的信任值进行比较,如果自己的信任值最大,该节点便进入 *ClusterHead* 状态,并立即通过发送 *CH* 消息告知所有邻居节点自己的状态转换;如果发现自己的信任值不是最大的,则继续等待直到收到 *CH* 消息,发送 *JoinMessage* 消息,得到簇首认证后转入 *ClusterMember* 状态,同时发送 *JOIN* 消息表明自己的状态转换。

1.5.2　*簇首维护*

簇首维护和节点维护同属于簇维护策略。若系统采用周期性更新簇首的方法来维护簇结构,当更新频率较高时,可以维护较准确的网络拓扑信息和较好的分簇结构,但是计算开销和能源耗费过大;而当更新频率较低时又会导致拓扑信息过时,甚至造成会话终止。为此,本文没有采用周期性簇维护策略,而采用更加灵活的动态簇维护机制即按需触发簇维护的策略。

簇首维护包括:簇首竞争,簇首离开/消亡及簇的合并。

簇首竞争　当簇内节点对簇首的信任评估值低于门限值时,通过在簇内发送申告消息来通知其他成员节点,当不同的成员节点发送申告消息的条数达到一定门限值,则在该簇内引发簇首竞争。

簇的合并　当两个簇的簇首移动至彼此邻居节点范围内,根据簇成员门限值判断是否引发簇首竞争。簇首竞争中成员少的簇的簇首放弃簇首状态。

簇首离开/消亡　簇首接到命令离开或因战场上各种复杂因素导致无法再执行簇首功能的情况。前者离开时发广播消息通知簇内成员,簇成员将节点状态转换为“未决定”后触发簇首竞争;后者当簇首消亡时无任何通告,簇成员在 n 个簇首消息广播周期内均未检测到簇首状态时转换状态为“未决定”触发簇首竞争。

1.5.3　*簇成员建立*

簇成员维护包括:节点入网,簇成员离开/消亡及节点簇间移动。

节点入网　节点第一次入网,此时节点为未决状态。若节点收到的簇首响应多于一条,则选择有隶属关系的簇加入。

节点离开/消亡　接到命令离开退网和非本意离开。前者广播其离开消息,簇首收到后删除其信息;后者经检测后发现其通信链路不可达后被簇首删除成员信息。

节点簇间移动　节点接到命令离开原簇到其他簇覆盖范围执行任务。

根据算法描述中节点的状态转移关系图,构建如下基于 Petri 网的可达树:

节点加入/离开簇的情况: $P_0P_1P_2P_3P_4P_5$ 分别代表节点的 6 个状态:未决定、等待、未认证、簇成员、簇首、离开。

t_0:发送 *Hello* 消息; t_1:没有收到 *CH* 响应消息;t_4:符合簇首条件; t_5:认证通过;t_6:收到离开指令。

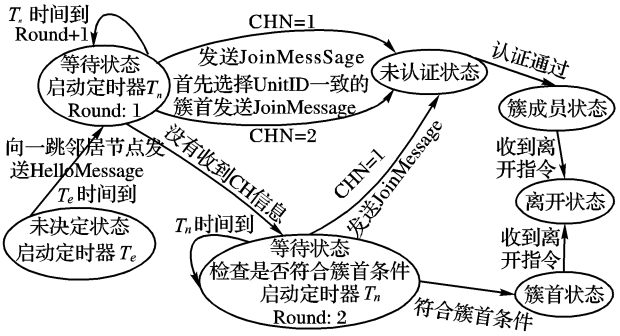

图 3　节点维护状态转换

通过可达树的分析,可对算法的各种性质得出结论:

1) 有界性。可达树中每个节点,每个位置中的标记数量都是有限的,所以此转换是有界的。

2) 活性。从可达树中可以看出,每个变迁至少被点火一次,不会出现死锁。

3) 完整性。在描述中所说的状态均可达。

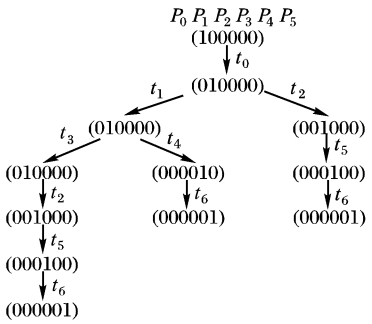

图 4　簇成员维护算法中节点状态转换可达树

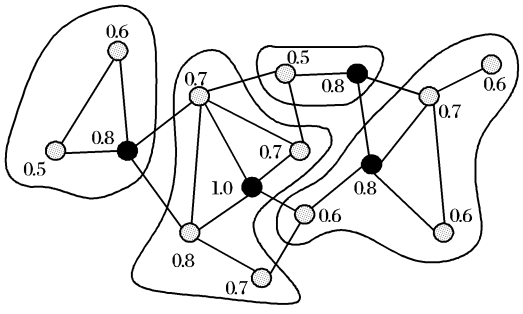

图 5　基于 SCABTE 算法的分簇

根据以上算法,可以构造以簇首为中心的一跳群,群内两个节点的最大距离不超过两跳。簇形成后,簇成员向簇首申请信任值证书。簇首在给簇成员发放信任值证书的同时发送自己的推荐证书用来证明自己的簇首身份是可信赖的,如图 5 所示,其中节点旁的数字为信任值,黑色节点为簇首。

2 算法安全性分析

下面从入侵节点的角度,分析算法的安全性:

1)恶意节点

在战术互联网中,簇生成消息均需要节点的签名,并且利用通信密钥加密。而恶意节点为非授权节点,无法获得通信密钥以及证书,因此恶意节点不可能通过伪造簇生成消息破坏簇生成过程。同理,恶意节点也不可能通过修改、重放等方式破坏簇首选举和维护过程。并且,在 SCABTE 算法中,消息均由通信密钥加密,因此恶意节点也不可能获得网络的拓扑信息。

2)合谋节点

由于战术互联网的节点物理安全性较差,在敌对环境中被捕获或接管的节点将会对网络造成极大的威胁。因此,簇生成算法必须提供检测并排除合谋节点的机制。在 SCABTE 算法中,合谋节点主要通过发送虚假的消息以及不遵守算法过程等方式破坏簇生成和维护。算法采用 Rakesh 不对称加密法,同时在消息中绑定数字签名和发送者的公钥。这样接受者就可以检测公钥是否是发送者的,从而断定消息是否伪造。

① 合谋节点无法伪造来自其他节点的推荐证书,而无法伪造虚假的 CH 消息,因而不能破坏簇首竞争过程。

② 合谋节点发送虚假的 JOIN 消息,仅影响合谋节点自身,使合谋节点反复加入退出簇,不会对簇生成的安全性造成影响。合谋节点簇间移动时向欲加入簇簇首伪造加入信息时,因其无法伪造原簇簇首发放的信任证书,而无法得逞,只以新入网的身份加入簇也不可能获得较高的信任值而破坏簇的安全。

③ 节点在接收到 NotifyMessage 消息后,首先验证消息是否来自授权节点及其完整性,如果证据信息可信,则将合谋节点的证书列入“黑名单”,从而将合谋节点排除在簇生成过程之外。

④ 合谋节点在簇合并过程中,始终不放弃簇首状态或仅接收消息不发送消息,对簇生成的影响较小,在此不作深入讨论。

3)懒惰节点

在 SCABTE 算法中,如果节点始终不决定自己的状态,也会对算法的安全性造成危害,可以将此类的合谋节点称为懒惰的合谋节点,如图 6 所示。

在图 6 中,如果懒惰节点 3 一直不确定自己的状态,则节点 1、2 和 4 一直等待 3 决定状态后才能确定自己的状态,节点 5 也需要等待节点 4 确定自己的状态,从而导致簇生成的过程始终不能完成。

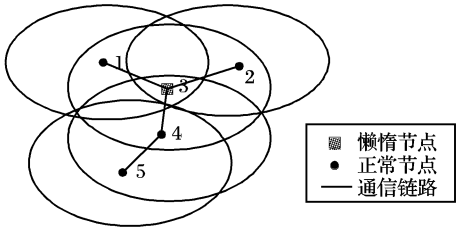

图 6 懒惰节点示例

对于懒惰节点,SCABTE 算法在 *HELLO* 消息中引入证据信息,说明节点无法确定自己状态的原因,即其邻居节点中仍存在处于未决状态且权值高于自己的节点。当节点 i 始终不确定自己的状态时,则 $HELLO_i$ 中需包含 $HELLO_j$,二者的时间差满足 $|curtime_i - curtime_j| < 2T_h$,且 $w_j > w_i$。若节点 i 为邻居节点集合中处于未决状态的权值最大的节点,则将周期广播 $HELLO_i$,且 $HELLO_j$ 为空;若节点 i 为懒惰节点,则始终广播 $HELLO_i$ 消息。在 SCABTE 算法中,若节点 v 连续接收到 $HELLO_i$,其中 $HELLO_j$ 为空,且消息的时间跨度满足 $|curtime_{first} - curtime_{end}| > T_w$,则将节点 i 标识为合谋节点,将 *HELLO* 消息作为证据信息广播 $NOTIFY(i, v)$ 消息。

同时,SCABTE 算法满足以下性质:

1)每个节点仅加入一个簇

在 SCABTE 算法中,节点通过发送 JOIN 消息加入相应的簇。仅当其在簇首通信范围之外,自己不符合簇首条件时,节点才转换到未决状态。由于每个节点的通信能力相同,不会存在节点在没有收到离开指令的情况下,转换到未决状态而后加入其他簇的情形。因此,在 SCABTE 算法中,每个节点仅加入一个簇。

2)不存在始终处于未决状态的节点

在算法中,一旦发现邻居节点存在簇首节点,节点将转换为簇成员状态。节点在所有信任值高于自己的邻居节点确定状态后,确定自己的状态。所以处于未决状态的节点不可能存在相互等待的情形。因此,算法中不存在始终处于未决状态的节点。

3 模拟实现与性能仿真

本文采用安装在 Linux 操作系统的 NS-2 模拟器[6]对算法进行模拟并分析其性能。模拟时需要设计一定的移动模型来反映动态的拓扑结构、链路的连接与断开。战术互联网中,节点由于作战使命及编制的约束,移动节点体现出“集团移动性”。本文采用最为接近战术互联网真实应用环境的参考点组移动模型(Reference Point Group Mobility Model, RPGMM)。每个组都有一个逻辑上的中心,每个移动节点都有自己的参考点。算法模拟参数如表 1 所示。

表 1 模拟参数表

参数	含义	默认值
T_h	节点间交换簇生成消息的周期	2 s
T_w	链路被认为是有效的最大时间	4.2 s
Tc	推举簇首等待时间	5 s
Node number	移动节点数目	500
Node number/group	每组节点数	50
Sim-time	模拟时间长度	600 s
reference point separation	参考点离散度	500 m
Pause time	RPGM 模型中节点停等时间	2 ~ 6 s
Length	区域长度	40 km
Width	区域宽度	40 km
Max-speed	节点最大移动速度	0 ~ 50 m/s
Tx-range	无线传输距离	150 ~ 600 m

算法的性能度量包括:

1)节点状态变化频率。单位时间内成员节点状态变化次数,用成员节点状态变化总数除以模拟时间表示。由于节点频繁移动,链路变化将导致节点状态变化。显然节点状态变化次数越少,簇结构越稳定。

2)簇首节点变化频率。单位时间内簇首节点变化次数,用簇首节点变化总数除以模拟时间表示。在战术互联网络中,因簇首节点邻接而触发簇首竞争导致簇首节点状态变化,相应的簇成员节点也随之变化,从而引起簇结构抖动。因此,簇生成算法的簇首节点变化频率越小说明生成的簇结构越稳定。

在已有的簇生成算法中,选择 Lowest-ID[7]算法以及 MIX 算法[8]与 SCABTE 算法进行性能比较,原因在于这些算法的

簇首选举标准易于验证。

1)算法随节点传输距离变化的情况

图 7 显示了在 RPGMM 模型下簇生成算法性能随无线传输距离变化的情况。当传输距离大于 16 km 后,簇首变化频率随着无线通信距离的增加而逐渐下降,原因在于组移动模型下,同组节点具有一定的聚合性。在参数设置时,同组节点距离参考点的平均距离为 500 m。此外,同组节点在移动时具有一定的方向性,因此同组节点的相对移动性较小。因此,当无线传输距离增加时,簇首数量迅速减少,簇首竞争概率下降。

从图 7 中可以看出,SCABTE 算法的性能明显优于其他算法,根据簇首选举标准,SCABTE 算法尽量将邻居节点组成一个簇结构,随着无线通信距离的增加,邻居节点组成一个簇的可能性越高。同时簇首竞争原则避免了两个组交错时导致簇结构的变化,因而具有较好的稳定性。

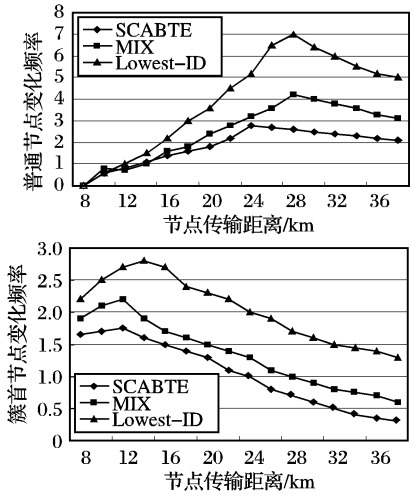

图 7　节点传输距离变化时算法的性能

2)算法随节点移动速度的变化速度情况

图 8 显示了参考点组移动模型下簇生成算法的性能随节点移动速度变化的情况。在 RPGMM 模型中,节点的移动速度越大,簇结构的稳定性越低,簇首节点变化频率与簇成员节点变化频率均随着节点频率移动速度的增加而提高。

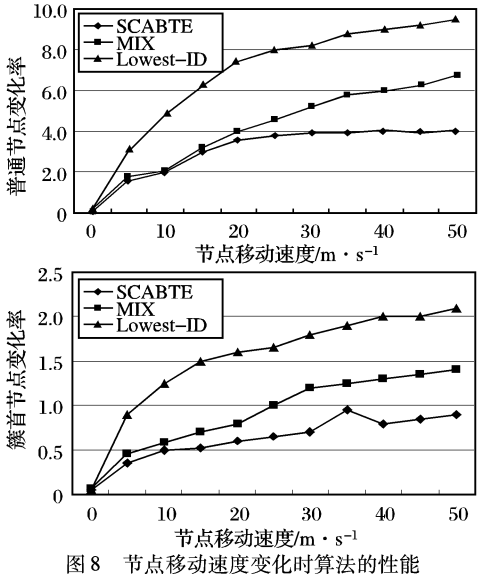

图 8　节点移动速度变化时算法的性能

然而,如图 8 所示 SCABTE 算法受节点移动速度的影响明显小于其他簇生成算法。这是因为簇首竞争原则有效地避免了节点高速移动导致的簇首频繁竞争,提高了簇结构的稳定性。

3)算法随网络规模变化的情况

当网络规模变化时,节点的移动速度为 15 m/s,通信能力为 15 km 时,从图 9 可以看出,节点状态变化频率和簇首节点变化频率与网络规模基本呈线性关系,其中 Lowest-ID 算法的增长速度最快,而 SCABTE 算法的变化频率小于其他算法且增长速度最慢。

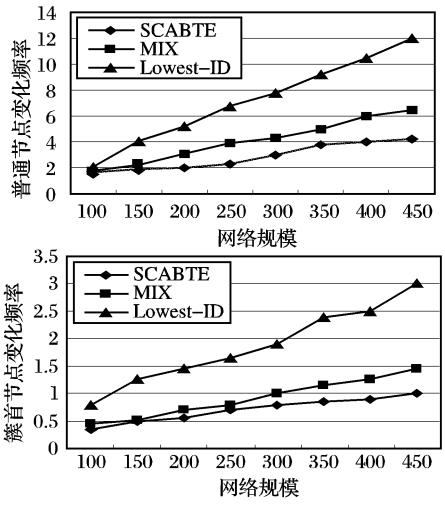

图 9　网络规模变化时算法的性能

通过模拟分析可以看出,SCABTE 算法在节点通信能力、移动速度以及网络规模变化的情况下,簇结构的稳定性均明显优于 MIX 和 Lowest-ID 算法。虽然算法的簇首数量的最小值较 MIX 算法大,但是簇首数量随模拟时间的变化幅度较小。因此,SCABTE 算法的性能明显优于其他簇生成算法,适合为战术互联网络建立基于簇的层次结构。

4　结语

战术互联网同质网层特性说明引入分簇的必要性和优势,针对已有分簇算法安全方面的欠缺,结合信任评估,信任关系建立的思想提出了一种适用于构建战术互联网同质网层分级结构的安全分簇算法。从入侵节点的角度分析了其安全性。最后用 NS-2 构建了战术互联网仿真环境,在节点传输距离、节点移动速度、网络规模变化条件下验证了算法的可用性。

参考文献:

[1] KELSCH G R. A common tactical internet performance model architecture[C]// MILCOM'97 Conference Proceedings, Nov. 2 - 5, 1997. [S.l.]: IEEE, 1997, 1: 177 - 181.

[2] XU K X, HONG X Y. An ad hoc network with mobile backbones [C]// Proceedings of IEEE International Conference on Communications (ICC 2002), New York, NY, April 2002. New York, NY: IEEE Computer Society Press, 2002: 3318 - 3324.

[3] RUBIN I, VINCENT P. Topological synthesis of mobile backbone networks for managing ad hoc wireless networks[C]// MMNS2001, LNCS2206. London, UK: Springer-Verlag, 2001: 215 - 221.

[4] KONG J J, ZERFOS P, LUO H Y, *et al*. Providing robust and ubiquitous security support for mobile ad-hoc networks[C]// IEEE 9th International Conference on Network Protocols (ICNP'01), Riverside, California, November 11 - 14, 2001. [S.l.]: IEEE Computer Society, 2001: 251 - 260.

[5] 况晓辉, 朱培栋, 卢锡城. 移动自组网络分布式组密钥更新算法[J]. 软件学报, 2004, 15(5): 757 - 766.

[6] The University of Southern California's Information Sciences Institute. The network simulator NS-2[R/OL]. [2007 - 05 - 10]. http://www.isi.edu/nsnam/ns/.

[7] GERLA M, TZU - CHIEH TSAI J. Multicluster mobile multimedia radio network[J]. Wireless Networks, 1995, 1(3): 255 - 265.

[8] LIN C R, GERLA M. Adaptive clustering for mobile wireless networks[J]. IEEE Journal of Selected Areas in Communications, 1997, 15(6): 1265 - 1275.