基于攻击因子的 Ad hoc 网络多路径安全路由机制*)

黄 辰 王芙蓉 莫益军

(华中科技大学电子与信息工程系 武汉 430074)

摘 要 作为 Ad hoc 网络大规模应用的先决条件,安全问题必须得到解决。为了加强对中间节点的安全评估以及对多路径路由的保护,本文提出一种基于攻击因子 AF(Attacking Factor)的多路径安全路由机制,引入能够衡量节点当前及未来遭受攻击的可能性的攻击因子以评估多路径集的安全性,并在此基础上实现了链路安全预警,降低了恶意攻击带来的损失。仿真结果表明,在不安全的网络环境下,本文机制有效保护了路由控制消息,提高了多路径路由建立的效率。

关键词 Ad hoc,多路径,安全,路由

Security Multipath Routing Mechanism of Ad hoc Network Based on Attacking Factor

HUANG Chen WANG Fu-rong MO Yi-jun

(Department of Electronic and Information, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract As a prerequisite for broad application of Ad hoc network, security problem must be solved. To enhance the security evaluation of intermediate nodes and the protection of multipath routes, we present a secure multipath routing mechanism based on Attacking Factor(AF), which can indicate the possibility of being attacked currently and in the future. AF can be used to evaluate the security of multipath sets, then link security alarming can be implemented to reduce the loss from malicious attack. The simulation results show that this mechanism can protect routing control message effectively and improve the efficiency of multipath routing establishment in unstable environment.

Keywords Ad hoc, Multipath, Security, Routing

1 引言

按照在源和目的间建立的路径的数目,Ad hoc 网络路由协议分为单路径路由协议以及多路径路由协议。单路径路由协议仅建立一条延时最短或者能耗最低的路径,而多路径路由协议不仅提供多条可用路径,且允许节点自主选择如何使用这些路径。换句话说,多路径路由是用多条好的路径来代替单条最好的路径。

多路径路由提供了容错、负载平衡、聚合带宽等多种功能,是当前 Ad hoc 路由协议的发展方向和研究热点。它充分发挥了节点的寻路能力,由多条路径同时完成源目的节点的交互,所以每条路径的安全性都应该得到保障,这就使得对多路径的保护难度要明显高于单路径。

目前多路径路由协议研究的不足主要是将注意力集中在产生最多数量的节点不相交路径上,忽视了对这些路径的安全性考察[1];或为了降低开销,只进行端到端的认证,没有对中间节点的合法性给予足够关注[2];还有的只关心基于多路径的数据传输安全,没有考虑到对多路径构造过程的保护[3]。本文提出基于攻击因子 AF(Attacking Factor)的多路径安全路由机制,依据对节点遭受恶意攻击的可能性的评估,建立对多路径集安全性的量化模型,便于源节点选择符合自己安全需求的节点不相交多路径集进行数据传输。对目前多路径安全路由协议中对多路径建立过程的保护不够,缺乏对中间节点的监控和对多路径集的安全性评估的问题进行了改进,提高了路由服务的安全性和可靠性。

本文其余部分安排如下:第 2 节介绍本文提出的基于攻击因子的多路径路由安全机制的设计思想;第 3 节描述攻击因子的数学模型及相应的路由流程设计;第 4 节是安全性分析;第 5 节是仿真及结果分析;最后是全文的总结。

2 多路径安全路由机制的总体设计思想

攻击因子 AF 是对节点遭受攻击的可能性度量,包括当前和未来被攻击的可能性,因此将攻击因子细分为攻击影响因子 AIF(Attacking Influence Factor)和攻击诱惑因子 ATF(Attacking Tempt Factor)。攻击影响因子依据节点性能下降程度,推断节点当前受攻击的可能性;攻击诱惑因子依据节点在网络中的表现及所处的地位,推断其未来受攻击的可能性。路由过程中应尽可能地避免将路由控制消息和数据路由到高攻击因子的节点上。攻击因子计算复杂度低,比起基于签名认证的公钥体系,更适合运行在能量有限的 Ad hoc 节点中。

基于以上思想,本文为节点设计了如图 1 所示的模块图。其中信息分析模块根据从信息采集模块得到的各邻居节点的运行状况信息计算出相应的 AF 值,并将结果交由信息维护模块保存;信息维护模块定期对周边节点 AF 值进行更新,并通过滑动窗口预测断链可能性;依据信息维护模块的信息,对

*)本文获国家自然科学基金(60572047),教育部新世纪优秀人才支持计划(NCET-06-0642)资助。黄 辰 博士研究生,主要研究方向为对等自组织网络;王芙蓉 教授,博士生导师,主要研究方向为计算机网络与下一代移动通信;莫益军 讲师,博士研究生,主要研究方向为对等自组网与信息安全。

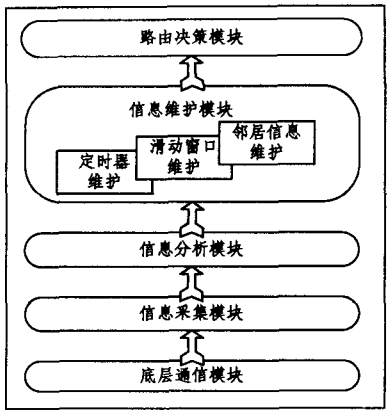

图1 基于攻击因子的模块设计图

路由控制信息的转发方向和多路径集的选择将由路由决策模块完成。因此本文是一套基于攻击因子的完整的 Ad hoc 安全多路径路由解决方案。

3 基于 AF 的多路径安全路由机制

3.1 攻击因子的数学模型

(1)攻击影响因子 AIF

节点受攻击后的性能变化主要有以下几方面：

· 节点吞吐量

节点的吞吐量指在不丢包的情况下能够接受的最大的包数量，用 N_{IO} 表示。吞吐量下降表明节点可能受到了 DoS 类攻击。

· 节点单任务处理时间

节点的单任务处理时间是平均处理一个消息请求所需的时间，用 T_{sig} 表示。如果单任务处理时间过长，表明可能受到了能量耗尽类型的攻击。

· 节点响应时间

节点响应时间指节点收到信息交换请求，即 Hello 消息后给予回应的时间，用 T_{resp} 表示。响应时间过长，表明可能已经被攻击者所控制，不能再参与到正常的路由进程中了。

以上 3 种性能的变化不一定完全代表节点受到攻击，可能是能量在运行中逐渐耗尽或者在网络中出现拥塞，因此本文定义 P_{IO}，P_{sig} 和 P_{resp} 分别表示 N_{IO}，T_{resp} 和 T_{resp} 3 种性能变化指代的当前受攻击概率，并得到 AIF 和当前受攻击概率的关系式：

$$AIF=\eta_1 * P_{IO}+\eta_2 * P_{sig}+\eta_3 * P_{resp} \quad (1)$$

其中 $\eta_1+\eta_2+\eta_3=1$，η_1，η_2，η_3 分别代表 3 种当前受攻击概率在 AIF 中所占的比重。

(2)攻击诱惑因子 ATF

节点未来遭受攻击的可能性依据的是当前节点在网络中所处的地位和发挥的作用。节点在网络中应尽量保持“低调”，避免自己成为网络运作的热点。一般来说，在以下特征参数上表现突出的节点，未来极有可能成为恶意攻击的目标：

· 节点利用率

节点利用率指单位时间内节点和周边邻居的交互次数，用 N_{util} 表示，这里的交互指的是路由控制消息的转发。

· 节点邻居数

节点的邻居数是指当前和节点进行信息交换的邻居节点的数目，用 N_n 表示。邻居数代表节点所在区域密度及邻居节点间产生路由交互的机率。

· 节点待处理的任务队列

节点待处理的任务队列表示节点当前正在等候处理的消息请求的数目，用 N_{quere} 表示，待处理队列的长度反映了该节点当前的拥塞程度。

N_{util} 越高，N_n 越多，N_{quere} 越长，意味节点网络活动频繁，攻击针对这样的热点能造成最大程度的破坏。同当前受攻击概率一样，定义 P_{util}，P_n，P_{quere} 分别表示 N_{util}，N_n 以及 N_{quere} 代表的未来受攻击概率，并得到 ATF 和未来受攻击概率的关系式：

$$ATF=\tau_1 * P_{util}+\tau_2 * P_n+\tau_3 * P_{quere} \quad (2)$$

其中 $\tau_1+\tau_2+\tau_3=1$，τ_1，τ_2，τ_3 分别代表 3 种节点未来受攻击概率在 ATF 中所占比重。

(3)攻击因子 AF

取得攻击影响因子 AIF 和攻击诱惑因子 ATF 后，可以得出节点的攻击因子 AF：

$$AF=\alpha * AIF+\beta * ATF \quad (3)$$

其中 $\alpha+\beta=1$，α，β 分别代表对于 AIF 以及 ATF 的置信度。

3.2 多路径安全路由机制的流程设计

基于 AF 的多路径安全路由机制分为路由请求、路由建立及路由维护 3 个阶段。本文设定危险阈值 AF_d，AF_d 是节点决定是否向邻居节点转发路由控制消息的衡量标准。AF 值高于 AF_d 的节点将被排除在多路径集的构建之外。

3.2.1 路由请求阶段

(1)源节点生成路由请求消息

本文在单路径安全路由协议 SAODV[4] 的 RREQ 消息中，附加 RouteList，ExcludeList，NextHop，AFList，AF Hash5 个字段，如图 2 所示。

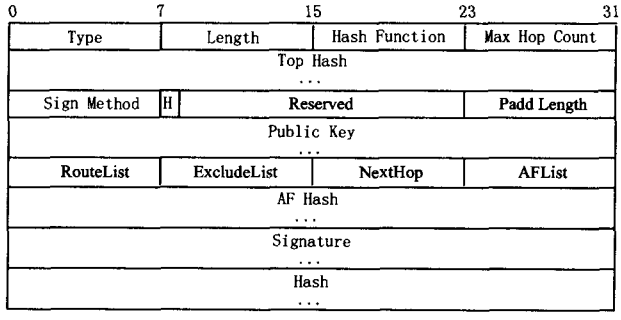

图2 基于 AF 的 RREQ 消息

RouteList：动态记录了经过的中间节点 ID；

ExcludeList：动态记录了被排除在路由进程之外的节点 ID；

NextHop：动态记录了下一跳的节点 ID；

AFList：动态记录了经过的中间节点的 AF 值；

AF Hash：用于验证 AFList 是否遭到篡改。

源节点生成秘密随机数 K_{sd} 作为 Hash 函数的种子，并用目的节点的公钥加密 K_{sd}，以保证其在传输过程中不被泄露。源节点计算：

$$H_{AF}=h(K_{sd},AF_N) \quad (4)$$

其中 AF_N 为选中的邻居节点的 AF 值，须低于 AF_d。H_{AF} 写入 AF Hash 字段中。RREQ 消息只会向 AF 值低于 AF_d 的邻居节点转发。

(2)中间节点处理路由请求消息

中间节点缓存最近处理的 RREQ 消息的摘要。摘要对象是源目的节点 ID、序列号以及 RouteList 字段，可以防止重

复处理且节省存储空间，相对于保存完整的 RREQ，摘要所需存储空间要小得多。

中间节点 n_i 收到 RREQ 消息后执行以下两部分算法：第一部分，检查 RREQ 消息有效性，通过检查后才缓存该 RREQ 消息摘要；第二部分，更新 RREQ 消息相关字段并转发。算法第一部分如下：

Step1　检查向自己发送 RREQ 消息的节点 AF 值是否超过了 AF_d。如果超过，丢弃该 RREQ 消息。

Step2　计算 RREQ 的摘要，同保存在缓存的摘要比较。如果有相同的，则丢弃该 RREQ 消息。

Step3　检查自己是否在 RREQ 消息的 NextHop 字段中。如果不在，丢弃该 RREQ 消息。

Step4　检查 RREQ 消息的 RouteList，ExcludeList 和 NextHop3 个字段中是否存在重复的节点，如果存在，丢弃该 RREQ 消息。

算法的第一部分的伪代码如下：

if($AF_{last} \geqslant AF_d$)

　　drop(RREQ)；

if(hashID_S，ID_D，Seq，RouteList)∈RouteCache)

　　drop(RREQ)；

if($ID_i \notin$ NextHop)

　　drop(RREQ)；

if((RouteList∩ExcludeList≠Φ)or

(RouteList∩ NextHop≠Φ) or (NextHop∩ ExcludeList≠Φ))

　　drop(RREQ)；

上述 4 步通过后，就可以将该 RREQ 消息的摘要加入节点缓存，继续进行算法第二部分：

Step5　挑选 AF 值低于 AF_d 的邻居节点，加入合格邻居集。

Step6　更新 RREQ 消息中相关字段：

a)将自己的 ID 附到原 RouteList 字段末尾；

b)将原 NextHop 字段中除自己以外的节点 ID 附到原 ExcludeList 字段的末尾；

c)用合格邻居集中的节点 ID 替代原 NextHop 字段，并删去和其它两个字段重复的节点 ID；

d)将新 NextHop 字段中的节点的 AF 值附在原 AFList 字段的末尾。

Step7　由原 AF Hash 字段及式(5)计算转发到不同邻居的 RREQ 消息中的 AF Hash 字段；

$$H_{AF}=h(H_{AF},A_{FN}) \tag{5}$$

算法第二部分，伪代码如下：

do{add(n_k，EligNeig)；}

while ($AF_k \leqslant AF_d$ and $n_k \in N_i$)；

RouteList=RouteList+ID_i；

ExcludeList=ExcludeList+(NextHop−ID_i)；

NextHop=N_i−(N_i∩RouteList)−(N_i∩ ExcludeList)；

AFList=AFList+AF_{Ni}；

do{H_{AF}=hash(H_{AF}，AF_k)；

　　AF Hash = H_{AF}；}

while($n_k \in$ EligSet)；

3.2.2　*路由建立阶段*

(1)目的节点生成节点不相交多路径集

目的节点每收到一条 RREQ 消息，根据式(6)递归计算出 H'_{AF} 并和消息中携带的 H_{AF} 比较，如果 $H'_{AF}=H_{AF}$，证明 AFList 在传输的过程没有被篡改；

$$\begin{cases} H_{AF}'=\text{hash}(K_{sd},AF_1) \\ H_{AF}'=\text{hash}(H'_{AF},AF_2) \\ \cdots \\ H_{AF}'=\text{hash}(H'_{AF},AF_D) \end{cases} \tag{6}$$

假设目的节点收到 m 条真实有效的 RREQ 消息 $RREQ_1,\cdots,RREQ_m$ 并提取出 $RouteList_1,\cdots,RouteList_m$ 构造节点不相交的多路径集。设 k 条路径节点不相交，节点不相交多路径集可以表示为

$$\text{DisjointSet}=\{\text{Routelist}_1,\cdots,\text{Routlist}_m \mid \bigcap_{i=1}^{k}\text{Routelist}_i=\varnothing\}$$

(2)目的节点生成路由回复消息

目的节点生成秘密随机数 K_{ds} 作为 Hash 函数的种子，并用源节点的公钥加密。目的节点计算：

$$H_{AF}=\{K_{ds},AF_N\} \tag{7}$$

H_{AF} 写入路由回复消息 RREP 的 AF Hash 字段。DisjointSet 的 RouteList 写入 RouteList 字段，RREP 消息只沿着 RouteList 中的节点传输。

为将多路径集信息安全完整发送给源节点，本文采用(n，k)门限秘密共享[5]将多路径集信息分为 k 份，由多路径集中的 k 条路径将子信息传给源节点。只有集齐 n 份，才能恢复出原始的多路径信息。

(3)中间节点处理路由回复消息

中间节点 n_i 收到 RREP 消息后执行以下算法：

Step1　检查上一跳节点 AF 值是否超过了 AF_d。如果超过，丢弃该 RREP 消息。

Step2　检查自己是否在 RREP 消息的 RouteList 字段中。如果不在，丢弃该 RREP 消息。

Step3　检查 RREP 消息中 RouteList 字段中自己的上一跳和下一跳节点是否属于自己的邻居节点。如果不是，丢弃该 RREP 消息。

Step4　将 RouteList 字段中下一跳节点当前 AF 值附在 AFList 字段相应节点原 AF 的后面，并计算相应的 AF Hash。

(4)源节点路由选择

RREP 消息的 AF List 字段中每个节点都有其上一跳和下一跳对其进行 AF 评价，上一跳给出的 AF_{pre} 在路由查找阶段，下一跳给出的 AF_{next} 在路由建立阶段。为防止恶意节点提高其它节点的 AF 值，源节点根据式(8)决定中间节点的最终 AF 值：

$$AF=\begin{cases} AF_{next} & AF_{next} \leqslant AF_{pre} \\ \dfrac{AF_{next}+AF_{pre}}{2} & AF_{next} > AF_{pre} \end{cases} \tag{8}$$

为描述路径及多路径集的安全性，定义 $P_{route}(N)$ 为拥有 N 个节点(不包括源节点)的路径受攻击的可能性度量，$P_{set}(M)$ 为拥有 M 条路径的多路径集受攻击的可能性度量。

如果路径表示为 Route = {S，n_1，n_2，…，D}，则

$$P_{route}(N)=1-(1-AF_1)\cdots(1-AF_D) \tag{9}$$

如果节点不相交多路径集表示为 DisjointSet = {$Route_1$，…，$Route_M$}，则

$$P_{set}(M)=\prod_{i=1}^{M}P_{route\,i}(N) \tag{10}$$

源节点对于多路径集的安全需求为 τ_{set}，即多路径集的

$P_{set}(M)$ 不超过 τ_{set}。以 $DisjointSet_1$ 为例，$DisjointSet_1 = \{Route_1, \cdots, Route_M\}$。计算其中各条路径的 $P_{route}(N)$，不妨设 $P_{route1}(N) \leqslant P_{route2}(N) \leqslant \cdots \leqslant P_{route3}(N)$，并计算 $DisjiontSet_1$ 的 $P_{set}(M)$。

源节点将执行以下算法：

Step1　比较 $P_{set}(M)$ 和 τ_{set}，如果 $P_{set}(M) > \tau_{set}$，则去掉 $P_{route}(N)$ 最高的路径 $Route_M$，计算新 $P_{set}(M)$。

Step2　重复 Step1，直至 $P_{set}(M) < \tau_{set}$，并将 $DisjiontSet_1$ 加入到合格多路径集中。

Step3　对其它多路径集重复 Step1 和 Step2。

源节点从合格多路径集中选择 $P_{set}(M)$ 最小多路径集，作为源目的节点间的数据通道。源节点根据自身要求选择多路径集，既保证了源节点自主选择路径的能力，同时使得即便多路径信息在中途被截取，攻击者也不能以此确定最终数据传输通道。

3.2.3　路由维护阶段

本文引入滑动窗口监控周边节点。滑动窗口从低到高保存邻居节点的 AF 值，其中头部 AF 值最低。如果新 AF 值低于头部 AF 值，则将新 AF 值放入滑动窗口头部，并清空后面窗口；如果新 AF 值高于尾部 AF 值，则将新 AF 值保存到滑动窗口尾部，且所有窗口向前滑动一个窗口，原头部被移出。

一旦 AF 值高于危险阈值 AF_d，则与该邻居间的链路就被标记为危险，但不必发起路由维护。当接下来收到的该节点的 AF 值仍高于 AF_d 且呈上升趋势时才会发出 RRER 消息。这样，保证抢先发起路由维护进程，最大限度地降低了误报的可能。

4　安全性分析

(1)邻居节点间的监控

在公钥体系中，邻居节点的合法性需通过验证其签名来完成。一旦邻居节点被劫持，攻击者将一并获取其私钥，这样对私钥签名的验证就失去了意义。而在基于攻击因子的本地监控环境下，节点可通过周期性和邻居节点间的信息交换，获取周边节点当前的安全状况，同时完成入侵检测。节点只需简单地比较节点当前的 AF 值和威胁阈值 AF_d，即可决定路由控制消息的发送方向。AF 值高于 AF_d 的节点不能参与路由进程。

(2)AF 的不可篡改性

本文提出的基于 AF 的多路径安全路由机制的关键是 AF 值不能被恶意篡改。一旦 AFList 中的 AF 值不能真实反映节点受恶意攻击的可能性，就极易造成误判。因此本文利用了 AF Hash 字段和单向哈希函数的不可逆性保护 AFList。路由控制消息在收集沿路节点对下一跳节点的 AF 评价的同时，会根据式(6)更新 AF Hash 字段。一旦下一跳节点更改了对自己的 AF 评价，最后就会被收到路由控制消息的源目的节点根据式(7)检验出来。

(3)多路径信息传输的安全

采用(n,k)门限秘密共享发送多路径信息，提高了其在传输过程中的安全性。路径集中的各条路径在传输的过程中是相互独立的，攻击者必须截取到至少 n 条路径上的子信息才能恢复出原始的多路径信息，这就大大增加了攻击难度。并且，通过一定的冗余，允许在 $k-n$ 条路径失效的情况下，源节点仍然能够依据余下的 n 条路径携带的子信息恢复出原始的多路径信息。

5　仿真

在仿真平台 NS2 上对目前应用广泛的多路径路由协议 AOMDV[6] 和本文提出的基于 AF 的多路径安全路由机制进行仿真和性能比较。仿真环境如下：MAC 协议采用 IEEE802.11b，仿真节点数为 500，节点在边长为 2000m 的正方形区域内随机分布且自由移动，移动速度为 3m/s，带宽为 10M。网络中 10%的节点为恶意攻击节点。

仿真中主要对 4 种性能指标进行了评价：

①成功投递率。是成功到达目的的数据包数量与发送总数之比，评价协议安全路由能力；

②平均吞吐量。是单位时间经过网络传输的数据包流量大小，该指标可以衡量协议在网络发生波动情况下的性能优劣；

③平均路由控制消息开销。是生成一个多路径集平均所需的 RREQ 消息的数量，该指标衡量了路由查找过程带来的网络通信开销；

④平均端到端时延。指从源节点发出 RREQ 消息开始，至收到 RREP 消息为止经历的时间，该指标衡量了协议构建多路径路由的速度。

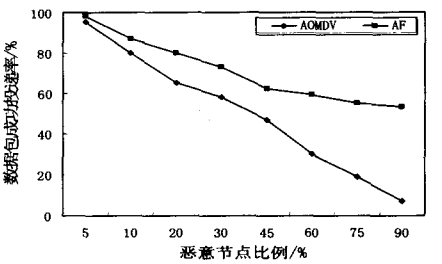

图 3　数据包成功投递率的比较

图 3 模拟恶意节点比例逐渐提高情况下 AOMDV 和基于 AF 的安全路由机制在成功投递率上的比较。仿真结果显示，基于 AF 的安全路由机制在恶意节点比例占到 90%时，成功投递率超过 50%，而 AOMDV 基本丧失路由能力。这是由于基于 AF 的安全路由机制将 AF 值超过威胁阈值的节点排除在路由进程之外，通过路径安全性评估选择受攻击威胁最小的路径加入多路径集，因此受攻击的影响较小。

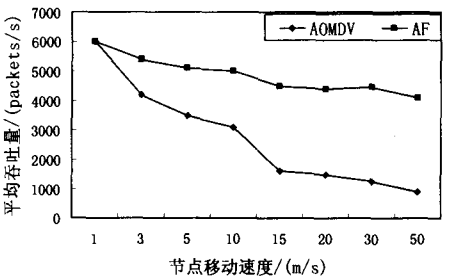

图 4　平均吞吐量的比较

图 4 模拟节点速度逐渐提高的情况下，AOMDV 和基于 AF 的多路径安全路由机制在平均吞吐量上的比较。仿真结果显示，随着节点移动速度的提高，断链现象频繁发生。AOMDV 在发现断链后才进行路由切换，致使大量数据包丢失。而基于 AF 的多路径安全路由机制有链路安全预警，能

(下转第 82 页)

式下的协议执行序列满足其公平性，在遵守型模式和可信任的背离模式下的证明过程基本相同，证明简洁易懂。然而我们的分析是建立在底层密码协议提供信道安全的基础之上，密码协议的前提是参与者必须诚实，但是电子商务协议中参与者却可能不诚实，因此采用协议分层的方法来验证电子商务协议属性还有待进一步研究。

参 考 文 献

[1] Syverson P F, Cervesato I. The logic of authentication protocols // Focardi R, Gorrieri R, eds. Foundations of Security Analysis and Design. volume LNCS 2171, Springer-Verlag, 2001

[2] Zhou Jianying, Gollmann D. Towards verification of non-repudiation protocols // Vickers T, Grundy J, Schwenke M, eds. International Refinement Workshop and Formal Methods Pacific 1998. Springer-Verlag, 1998: 370-380

[3] Wong H-C. Protecting Individuals' Interests in Electronic Commerce Protocols [D]. Computer Science Deptartment, Carnegie Mellon University, 2000

[4] Bella G, Paulson L C. Mechanical proofs about a non-repudiation protocol // Boulton R J, Jackson P B, eds. Theorem Proving in Higher Order Logics: TPHOLs 2001, LNCS 2152. Springer, 2001: 91-104

[5] Longo C, Bella G, Paulson L C. Verifying second-level security protocols // 16th International Conference on Theorem Proving in Higher Order Logics, volume LNCS 2758. Springer Verlag, 2003: 352-366

[6] Zhou Jianying, Gollmann D. An Efficient Non-repudiation Protocol // Proceedings of the 1997 IEEE Computer Security Foundations Workshop (CSFW 10). IEEE CS Press, 1997: 126-132

[7] Schneider S. Formal Analysis of a Non-repudiation Protocol // Proceedings of the 11th IEEE Computer Security Foundations Workshop. 1998: 54

[8] Qing S H, Li G C. A formal model of fair exchange protocols [J]. Science in China Series F-Information Sciences, 2005, 48 (4): 499-512

[9] Zhang L, Yin J P, Long J. Formal Analysis of Fairness of the ZG protocol[C] // Proceeding of Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (IEEE SecPerU 2005)

[10] Zhang L, Yin J P, Li M J. Formal Analysis of NetBill and Improvement[C] // Proceedings of the SKLOIS Conference on Information Security and Cryptology (CISC2005). Higher Education Press, 2005: 287-296

[11] 卿斯汉. 安全协议的设计与逻辑分析. 软件学报, 2003, 14(7): 1300-1309

[12] 周典萃，卿斯汉，周展飞. 一种分析电子商务协议的新工具. 软件学报, 2001, 12(9): 1318-1328

(上接第 55 页)

够在断链之前抢先发起路由维护，将有断链威胁的路径上的数据包平稳切换到其它路径，使对吞吐量的影响最低。

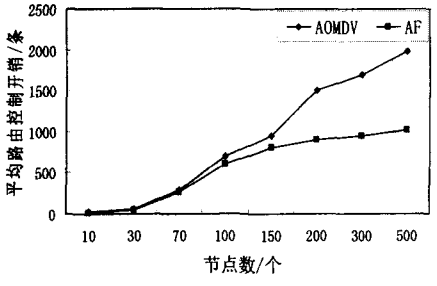

图 5 平均路由控制消息开销的比较

图 5 模拟节点数逐渐增加情况下，AOMDV 和基于 AF 的安全路由机制在平均路由控制消息开销上的比较。仿真结果显示，节点数达到 100 后，AOMDV 的 RREQ 消息数量会随节点数增加而大幅增加；而基于 AF 的安全路由机制由于有选择地向 AF 值低于安全阈值的周边节点转发 RREQ，并保证同一节点不会参与多条路径的形成，RREQ 消息数量增长较少。

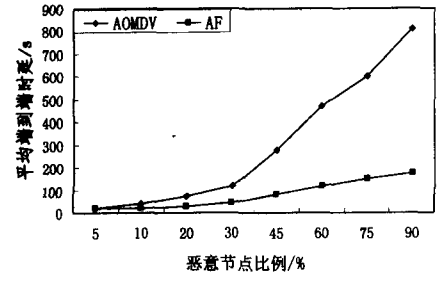

图 6 平均端到端时延的比较

图 6 模拟恶意节点比例逐渐升高情况下 AOMDV 和基于 AF 的安全路由机制在平均端到端时延上的比较。仿真结果显示，AOMDV 在攻击逐渐加剧时，大量路由控制消息在传输过程中丢失，导致端到端时延增加；基于 AF 的安全路由机制有效避开了有威胁的节点，保护了路由控制消息，即便恶意节点比例达到 90%，路由控制消息也只会经过那些安全节点，使得端到端时延保持在一个较低的水准上。

结束语 本文通过攻击因子 AF 衡量节点当前和未来遭受攻击的可能性，在此基础上建立了安全多路径集，并实现了抢先式路由维护。仿真实验证明，在基于 AF 的多路径安全路由机制的网络中，即使有相当比例的恶意节点存在，也不会对正常的路由进程产生太大影响。在保障多路径安全的同时，提高了路由建立的效率。将本文机制应用于动态环境下的大型网络中有较好的前景，值得进一步研究。

参 考 文 献

[1] Li Xuefei, Laurie G. Node-disjointness-based multipath routing for mobile ad hoc networks[C] // Proceedings of the 1st ACM International Workshop on PE-WASUN. 2004: 23-29

[2] Berton S, Yin H. Secure, Disjoint, Multipath Source Routing Protocol (SDMSR) for Mobile Ad Hoc Networks[C] // Proceedings of 5th Grid and Cooperative Computing (GCC' 2006). 2006: 387-394

[3] Mavropodi K. Secure Multipath Routing for Mobile Ad Hoc Networks[C] // Proceedings of Wireless On-demand Network Systems and Services (WONS'05). 2005: 89-96

[4] Zapata M G. Secure Ad hoc on-demand distance vector (SAODV) routing[Z]. Internet Draft, ddraft-guerrero-manet-saodv-06. txt, September 2006

[5] Kong Jiejun, Zerfos P, Luo Haiyun. Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks[C] // Proceedings of Ninth International Conference on Network Protocols. 2001: 251-260

[6] Marina M K, Das S R. On-demand Multipath Distance Vector Routing in Ad Hoc Networks[C] // Proceedings of the 1st International Conference for Network Protocols (ICNP). 2001: 14-23