移动自组网络中多径路由的匿名安全

章 洋 范植华 何晓新 徐帆江 王宇心
((中国科学院软件研究所通用软件实验室, 北京 100080)

摘 要: 多路径为移动自组网络提供的容错、负载均衡与 QoS 支持较单路径更有效可行, 所以在战术无线自组网等类似系统中采用多径路由策略更能满足系统的实际需求. 另外, 这类系统对安全性的要求除了基本的通信内容机密、完整与可用等特性外, 还要求通信者的身份与位置对敌人保密, 为通信者及其使命提供保护. 鉴于现有的移动自组网络的匿名路由协议都不是实用的多径路由协议, 且未能有效防御被动攻击、拜占庭行为以及匿名的不充分性, 本文设计了一种新型安全匿名的多径路由协议, 其特点是: 在移动自组网络中采用单私钥多公钥密码体制、Bloom Filter 与轻型洋葱盲化算法, 来实现通信者身份匿名、位置隐藏与路由不可追踪; 为源节点提供充分的路由信息, 基于充分的信息使用强化学习算法来提高系统抵御被动攻击与拜占庭攻击等路由安全攻击的能力, 并增强数据传输的可靠性. 通过仿真与分析, 显示了算法有较好的性能并达到了所定义的匿名安全要求.

关键词: 多径路由; 强化学习; Bloom Filter; 多公钥密码体制; 匿名性; 移动自组网络
中图分类号: TP309 文献标识码: A 文章编号: 0372-2112 (2005) 11-2022-09

Anonymous Secure Multipath Routing in Mobile Ad-Hoc Networks

ZHANG Yang, FAN Zhi-Hua, HE Xiao-Xin, XU Fan-Jiang, WANG Yu-Xin
(General Software Lab, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

Abstract: Multipath Routing is more suitable than single route for systems such as tactical radio networks because it can efficiently support fault-tolerance, load balancing and QoS. In addition, anonymity should be one important part of overall solution for those in which confidentiality, integrity and availability should be realized as basic security. Whereas those anonymous secure routing protocols known in MANET can't protect routing from passive attacks and byzantine behavior and can't provide mutilpath either, we adopt incomparable public keys cryptosystem, Bloom Filter, lightweight onion shuffle algorithm and reinforcement learning paradigm to design anonymous secure multipath routing protocol. We validate the effectiveness of our design using extensive simulation and detailedly analyze the security and anonymity as defined.

Key words: multipath routing; reinforcement learning; Bloom Filter; incomparable public keys cryptosystem; Anonymity; MANET

1 引言

移动自组网络由于其通信介质的开放性、拓扑的动态变化性及缺少中心管理点等固有特性, 易遭受安全攻击, 尤其在存在敌意对手的环境中, 安全问题变得更为严重. 战术无线自组网等类似系统的安全, 不但涉及到通信内容的机密性、完整性、鉴别与可用等安全特性, 还因为敌手跟踪通信路由会给路由端点的通信者本身及其所要完成的秘密使命造成威胁, 所以还涉及到路由的匿名安全性. 例如, 战场上指挥员与执行秘密任务的下属单位进行通信时, 希望系统能够提供匿名安全, 以保护指挥员不被敌人定位及秘密任务不被发现. 文献[1~3] 对移动自组网络的匿名路由问题进行了研究, 它们各有其优缺点.

文献[1, 2] 达到了各自设定的匿名要求, 但是它们对匿名的定义比较宽松, 所以达到的匿名性不够充分. 例如, 在文献[2] 中目的节点的身份泄露给了路由的中间节点, 文献[1] 中中间节点的身份泄漏给了目的结点. 另外, 两篇文献所提供的算法都有一个共同的缺点, 就是路由上的中间节点可以计算出到源节点的距离. 文献[3] 所设计的方法达到了较高的安全性与较好的匿名性, 但是它不适用于敌手能力较强的环境, 当路由中间节点被侵蚀而行为异常时, 即使只是简单地拒绝进行数据路由, 也难以检测与定位此节点, 从而不能有效维护路由与传输数据. 文献[2] 提出了使用多路径进行匿名传输的可能性, 但是没有深入探讨多径路由的性能、路径多寡及多径安

收稿日期: 2005-04-29; 修回日期: 2005-07-25
基金项目: 国家 863 高技术研究发展计划(No. 2005AA123110)

全性等问题, 未给出切实可行的方案. 文献[4~5]等也提供了多种移动自组网络安全路由协议, 但是这些解决方案不能直接应用于匿名路由, 无法以扩展原方案的简洁方式实现匿名性.

在移动自组网络中, 无线连接是不可靠的, 并且节点移动也导致网络拓扑结构持续变化, 因此单径路由失败的概率很高, 频繁启动路由发现过程会增加网络负荷, 过高的网络负荷进一步提升了路由失败的可能、降低了网络路由性能. 多路径由于有多条路径可以选择用来传输数据, 因此能提供容错、负载均衡、高的汇聚带宽及服务质量保证等功能特性. 由于二者路由能力的差异, 多径路由策略成了战术无线自组网等类似系统路由设计的首要选择.

文献[6~8, 13]提出了几种比较典型移动自组网络多径路由协议, 这些协议虽然建立了多路径, 但使用多路径的策略较简单, 常常是多条路径互为备用, 当主路径不可用时, 使用备用路径, 未能充分利用路由发现过程中获得的网络拓扑信息; 有些文献将重点放在对不相交路径的寻找上, 但这种路径上任意单跳链路出问题, 整个路径就失效, 代价很高. 文献[8]采用动态源路由的方法获得尽可能多的路由, 然后从中选择两条最大不相交的路径, 此方法为本文的多径策略设计提供了很好思路, 但需对文献[8]中的方法进行改进, 目的不再是获得两条最大不相交的路径, 而是获得源与目的节点间更多拓扑信息, 然后利用反馈学习的方法, 充分发挥多径的优势, 为抵御拜占庭攻击与被动路由攻击提供条件.

本文采用类似群集智慧的强化学习算法来进行多径的利用. 群集智慧是使用相互协作的一组分布式代理进行一系列学习来解决优化问题, 其灵感来源于蚁群寻找食物与巢之间的最短路径的协作方式. 在移动自组网络的环境中, 利用群集智慧模型解决路由问题的研究文献很多. 本文进行协议设计时采用文献[9]提供的强化学习算法, 它的特点是通过对数据分组的确认进行反馈学习获得源路由的方案, 而不依赖于中间节点的逐步路由选择, 提高了系统抵御已知拜占庭安全攻击的能力, 与文中算法相关的知识见文献[10].

本文主要工作是, 首先, 为移动自组网络设计了基于强化学习模型的动态多径源路由发现与利用协议, 增强了系统防止被动攻击与拜占庭行为的能力; 其次, 采用单私钥多公钥密码体制[11]创建伪名系统, 为路由节点命名伪名, 以伪名系统为基础, 结合 Bloom filter[12]机制, 实现身份隐藏; 再次, 在传输报文时, 采用轻型洋葱盲化算法随机化数据, 以使报文不能被追踪与实现位置机密; 最后对算法进行了分析与仿真.

2　基本模型与工具

2.1　匿名路由的定义

本文采用文献[3]中所用的概念, 从匿名路由所应达到的目标的角度, 阐述路由的匿名性, 把满足下面三个要求的路由称为匿名路由.

①身份机密性: 通信的发送方(称为源节点, 简称源)与接收方(称为目的节点, 简称目的)的真实身份除了通信双方无第三方知道, 通信双方也不知道路由节点的真实身份.

②位置机密性: 源与目的的位置不被其他节点知晓, 路由节点不能获得到源与目的的距离; 满足这个条件的称之为弱位置机密路由协议. 若再满足源与目的不知道到路由节点的距离, 称之为强位置机密路由协议. 本文设计的是满足第一个条件的路由协议.

③路由的匿名性: 不能通过追踪分组包来发现源与目的节点; 非路由上的第三方不能发现任何路由信息; 第三方难以推断源与目的之间的通信传输模式.

2.2　攻击模型与网络假设

我们将移动自组网络中对匿名多径路由安全的攻击分为如下两类:

①被动攻击, 主要指对报文传输的非法窃听与分析, 窃听者通过监听网络数据流来获得路由信息, 它不破坏现有的路由操作, 具有很强的隐蔽性; 另外也将路由中间节点拒绝提供指定的路由服务归为被动攻击, 例如攻击者让路由包通过而不让数据包通过.

②主动攻击, 指阻止传输与破坏数据, 包括复制、篡改与删除节点间所交换的数据以及耗尽资源阻塞数据流等.

敌手往往采取被动与主动混合的方法发起攻击. 敌手能力假设为可自由地窃听, 但无全局窃听能力, 并且只具有有限的计算能力与侵蚀节点的能力. 有限的侵蚀节点的能力指在同一个时间段之内只能侵蚀与控制部分少量节点. 在此解释一下拜占庭行为, 它指入网节点所做导致路由性能降低或路径被破坏的任何行为.

对于网络, 首先假设无线信道是对称与双向的, 且信道分配是公平的; 在单跳间通信时, 敌手无法在接收方收到报文之前, 将自己刚收到的报文修改后再发送给接收者. 其次, 假设节点有较强的计算能力与存储能力, 可执行公钥算法, 例如野战指挥网络的节点. 最后, 假设源与目的是互信的, 共享秘密数据; 源与目的间端到端的上层安全机制不作研究; 对链路层与物理层的安全问题也不作考虑, 只假设相邻节点间进行链路层通信时能验证对方设备具有入网权限, 但不需验证对方的真实身份, 例如可通过向对方证明拥有某种秘密获得中继传输服务; 广播报文分组时, 使用普通而不是广播 MAC 地址作为源地址.

2.3　多公钥密码体制

本文采用文献[11]中提出的不可比较公钥体制, 为源节点创建多个伪名, 并在此基础上生成路由节点的伪名. 采用这种体制的优点是, 代表源节点的多个伪名之间不可比较, 无从发现伪名之间的联系; 另外当与不同目的节点通信而导致伪名个数增长时, 不加重源节点的密钥存储与管理负担. 本文将这种密码体制称为单私钥多公钥的密码体制, 以下简称多公钥密码体制. 下面简单介绍文献[11]中提供的基于 ElGamal 密码系统的多公钥生成算法:

给定一个安全参数, 生成一个大素数 p 作为全局参数, 使 $\frac{p-1}{2}$ 也是一个大素数; 以 p 为参数随机生成私钥 a; 从 Z_p^* 中随机选择一个二次剩余数 g, 计算 g^a, 输出公钥 (g, g^a); 这样的公钥可以生成多个, 且对应同一个私钥 a. 由于计算及通

信代价的原因, 本设计中并不直接使用此公钥加密数据.

2.4 Bloom Filter

Bloom filter 是一个 m 个比特位的向量 BF, 是对集合中 n 个元素的编码. 选择 k 个独立的散列函数, 每个都将集合中的元素映射为 $\{1, 2, \cdots, n\}$ 中的数. 开始将 BF 置零, 然后对集合中每个元素, 用 k 个函数散列, 得 k 个位置值, 将 BF 中这些位置上的比特值置 1.

要检验一个元素是否属于集合, 将其经过 k 个函数散列, 得到 k 个值, 若 BF 中相应位置都是 1, 则认为其属于集合. 此方法会把非集合中元素错判为集合中元素, 其概率为 $(1-(1-1/m)^{kn})^k$. 给定 m 与 n, 最优 k 值为 $\ln 2\times m/n$. 应用时需权衡 m、n、k. 可参看文献[12]. 在设计的协议中, 用此方法来显著节省带宽、减少存储空间与提高查询效率.

2.5 强化学习算法

强化学习是一类无导师学习, 即通过试验及其反馈来优化系统与动态环境的交互, 一般用马尔可夫决策过程作为强化学习问题的基本模型. 文中使用了类似于群集智慧的强化学习方法, 所用到的概念与公式均来自文献[9], 下面对文献[9]中的算法作一简介.

每个源节点维护自己的网络拓扑以及到达目的节点的概率分布. 源节点利用文献[9]中 B.1 节的算法将网络图变换成分层有向无环图; 用数据分组的确认报文建立反馈机制, 使图上每个节点能获得到邻近节点的传输成败情况; 用 B.2 与 B.3 节的公式对反馈结果进行计算, 得出节点入边的概率; 再用 B.4 节的公式得出节点出边概率; 为使决策正确, 以小概率随机对较少用到的边用数据分组进行采样, 用 B.5 节的公式计算采样的传输路径.

2.6 文章的基本思路

本文对文献[8]的按需多路由发现方法进行了性能与拓扑信息获取的改进. 首先中间节点只在第一次收到路由请求报文时向外广播, 对后续的路由请求, 若来源于不同的相邻节点而且距源节点的步跳数不大于第一次收到的, 则进行记录, 否则直接丢弃此请求. 其次, 路由信息不传到目的结点, 目的节点的路由应答次数是一个可配置的值. 最后, 路由中间节点不是简单地沿路由请求路径反向传输路由应答, 而是使用时间窗机制; 对在规定的时间窗内收到的路由应答报文, 节点向未曾收到应答的前驱节点中的一个转发应答, 若前驱节点都收到过应答则向任意一个前驱节点转发应答; 当时间窗口关闭时, 启动自动路由应答, 条件是仍有前驱节点未能接收到路由应答报文, 且本节点有通向目的节点的路径, 则向满足条件的前驱节点发送路由应答报文; 源节点据处理能力决定处理多少个路由应答. 此方法可能升高系统的路由负担, 降低网络性能, 但通过反馈学习可充分利用获得的拓扑信息, 减少由于链路失效及敌手攻击而导致发起路由发现过程的次数, 从而保持系统的总体路由性能.

源节点获得到目的结点多路径后, 形成网络拓扑图, 能从图上计算出新路径, 新路径可能不同于应答过程中返回的任何一条路径. 另外, 源节点需对每条链路进行评估、计算概率分布, 主要是通过对数据分组使用两组确认报文作为反馈机制完成的; 确认报文返回的第二路径途径使用较少的链路, 确认报文的主返回路径是沿数据分组传输路径的反向返回.

为了使前次会话过程的链路评估信息可被用于本轮会话的强化学习算法中, 在使用伪名机制时, 采用 Bloom filter 在源节点将中间节点的两次伪名联系起来; 这样做的优点是源节点可在不知道中间节点的真实身份情况下利用节点传输成败的历史信息. 对于其他节点而言, 它无法发现此中间节点的两个伪名间的联系, 推断不出其他相关信息.

身份匿名主要是通过基于多公钥密码体制的伪名机制来达到的, 不过所产生的伪名都是与特定的源与目的节点相关的, 且只用于当前会话. 在每个节点对报文相关部分进行随机盲化让输入与输出的报文完全不同, 使攻击者不能比较内容与计算特征量来追踪路由. 洋葱加密的方法虽能取得盲化效果, 但对大量数据的加密会造成节点过重计算负荷, 我们采用计算量更小的散列与异或运算来对数据进行洋葱式层叠盲化.

3 安全匿名的多径路由协议

在每个需要发送数据的源节点上启动路由发现过程与执行强化学习算法, 实际的数据分组是源路由发送的. 系统使用三种报文进行路由发现与多径利用: 路由请求报文、路由应答报文与数据分组的确认报文(以下简称确认报文). 路由请求报文通过广播完成, 可能多径到达目的节点; 路由应答报文沿路由请求报文传输路径的反方向多径返回; 确认报文沿数据传输的主路径与源节点选择的第二路径返回.

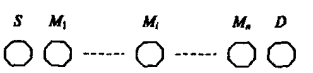

图 1 从源节点 S 到目的节点的路由

将源节点表示为 S, 中间节点表示为 M_i, 目的节点表示为 D, 也可把 S 看作 M_0, D 看作 M_{n+1}, 源与目的之间的某一特定路径的中间节点个数用 n 表示, 如图 1 所示.

将本文中常用的符号说明如下: $K(\,)$ 为使用对称密钥的加密函数, K 为对称密钥; $\{\}_{PK}$ 为使用公钥加密的函数, PK 为公钥; $H(\,)$ 表示标准的密码学散列函数, 例如 *SHA*-1、*RIPEMD*-160 与 *MD5* 等, $H1(\,)$ 与 $H2(\,)$ 表示不同标准的散列函数; $h(\,)$ 为一般散列函数; g 为大素数阶的群元素, 例如 2.3 节 Z_p^* 中的二次剩余数或椭圆曲线加法点群中的一个基点; g^x 表示指数运算, 例如乘法群中元素 g 的 x 次幂或点群中点 g 的 x 次倍点; *HOP* 表示步跳数; *TM* 表示时段; *RD* 表示随机数; *BF* 表示 Bloom filter; Len 表示变量的比特长度; ⊕表示按位异或运算; ⊙表示按位与运算; ≪表示左移运算; ≫表示右移运算; 〈〈表示循环左移; 〉〉表示循环右移; | 表示位串连接操作; Sizeof( ) 是求位串长度的函数; ZERO 表示所在的位置的比特值取二进制零, 0 表示所在的位置取随机数.

3.1 路由请求

源节点在如下两种情况下发起路由请求: 目的节点无路径可达; 与目的节点开始一个新的会话. 路由请求的设计类似于文献[3]的方法, 关键的不同之处是本设计中源节点与中间

节点采用伪名的方法进行身份隐藏, 而非完全消除名称. 路由请求报文数据格式如下:

$\lfloor RREQ,\ seq,\ g^a,\ g,\ RD_s,\ TM,\ K_{SD}(D,\ U_0,\ K_t),\ \{H_h,\ END\}_{PK_t},\ g^{x_{i-1}},\ U_{i-1}\rfloor$

其中, seq 此路由请求报文的序列号; g^a 源节点在发起路由请求时向外广播的伪名, 是其公钥(g, g^a)的一部分, 须不同于曾使用过的伪名; g 与 g^a 一起组成源节点公钥; RD_s 表示源节点发送的一个随机数; K_{SD} 源与目的之间共享的秘钥; D 目的节点身份标识; $H_h = H(seq, g^a, g, RD_s, TM)$; PK_t 为一次性公钥, K_t 与 PK_t 相应的一次性私钥; END 表示目的节点收到路由请求报文的一个标志数; $g^{x_{i-1}}$ 节点 M_{i-1} 的伪名, 用于在相邻节点间或与源节点通信时标识节点 M_{i-1}; U_0 源节点选择的随机数, 计算相对步跳; U_{i-1} 由节点 M_{i-1} 计算出的 U 数值; TM 此路由最大有效时间, 远大于会话时间.

每个中间节点的路由表项的关键数据结构如下:

```
struct RoutingEntry //路由表项
{seq, g, g^a;// 路由请求报文中的序列号与源节点公钥, 标识本次会话路由
RD_s, TM, {H_h, END}_PK_t;// 路由请求报文中相应数据项;
x, g^x, g^αx;// 本节点随机选择的秘密数、伪名及与源共享的秘密数;
stack PseudNodeS        //路由堆栈
  {g_last;   //前驱节点的伪名;
  g_next;   //后继节点的伪名}
  U;        //收到的第一个路由请求报文中的 U_{i-1}
  CreateTime;       //创建时间
}
```

在每个节点 M_i 上对 $U_i(i = 0, 1, \cdots, n)$ 的计算同文献[3], 如下:

$$U_i = (U_{i-1} \oplus R_i) \gg\gg p,\ i = 1, 2, \cdots, n \qquad (1)$$

R_i 是 M_i 选择的随机数, 其长度为 $p = Sizeof(U_i)/(HOP_{max} + 1)$, HOP_{max} 为路由的最大步跳数. 当 M_i 接收到多个路由请求报文时, 若这些报文经过的步跳数不大于 HOP_{max}, 可通过比较相互间 U_{i-1} 值的最低 p 位在对方 U_{i-1} 上的位置, 来确定到源节点距离的相对远近.

当路由请求报文到达一个节点时, 此节点查看路由表中是否有序列号为 seq 且源节点公钥为(g, g^a)的表项, 如果没有, 节点试图解密 $K_{SD}(D, U_0, K_t)$, 如果失败, 表明本节点不是目的节点, 而是中间节点. 中间节点 M_i 上处理过程如下:

(1)在路由表中添加新路由表项, 以 seq、g、g^a 为标识, CreateTime 为当前时间;

(2)从路由请求报文中提取 RD_s、$\{H_h, END\}_{PK_t}$、TM 对应填入路由表项;

(3) $U = U_{i-1}$, 并将 $g_{last} = g^{x_{i-1}}$ 压入堆栈;

(4)随机选择 x, 计算 g^x, 作为本节点伪名;

(5)如式(1)计算 U_i;

(6)修改路由请求报文, 令其中: $U_{i-1} = U_i$, $g^{x_{i-1}} = g^x$;

(7)向外广播修改过的路由请求报文.

若此节点超时未能收到路由应答报文, 则删除此表项, 释放资源. 对已经建立的路由表项的删除在余下两种情况下也需执行: 当前时间减去创建时间大于 TM; 收到源节点的清除路由信息的广播报文, 此报文经过源节点签名.

如果解密成功, 则表明本节点是目的节点. 首先用 K_t 解密$\{H_h, END\}_{PK_t}$, 验证 seq, g^a, g, RD_s, TM 及 END 合法性, 如不合法则终止处理. 其次, 目的节点可通过比较 U_0 与 U_{i-1} 提取路由长度, 如果路径超过最大值 HOP_{max}, 则终止处理. 最后记录相关数据, 并保存路由长度信息. 处理完毕后, 目的节点创建路由应答报文, 向外发送.

如果存在相应的路由表项, 则表明其他路经的存在. 中间节点首先检验是否应丢弃收到的路由请求报文, 提取报文中的 U_{i-1} 与 $g^{x_{i-1}}$, 在路由堆栈中查找是否有 g_{last} 等于 $g^{x_{i-1}}$, 如果存在, 则丢弃此路由请求报文; 否则将 U_{i-1} 与路由表项中的 U 比较, 如果 U 的最低 p 位不能在 U_{i-1} 上找到, 则丢弃此路由请求报文; 如果路由表项中记录的第二项数据与报文中相应数据不一致也同样终止处理. 其次, 将 $g_{last} = g^{x_{i-1}}$ 压入堆栈.

如果本节点为目的节点且重复收到路由请求报文, 验证 RD_s、TM、$K_{SD}(D, U_0, K_t)$、$\{H_h, END\}_{PK_t}$ 与记录中数据的一致性, 如不一致则终止处理; 若 $g^{x_{i-1}}$ 已存在, 也终止处理. 通过比较 U_0 与 U_{i-1} 值提取路由长度, 如果路由长度超过最大值则终止处理. 若计算出的路由长度小于记录中的路由长度, 则保存新路由; 否则检查记录中的路径数是否超过给定的域值, 若未超过, 则保存新路由, 反之将其丢弃. 对被接受的新路由, 创建路由应答报文.

3.2　路由应答

目的节点在接收到路由请求报文并验证合法后创建路由应答报文, 向外发送. 节点 M_i 收到的路由应答报文数据格式如下:

$\lfloor RREP,\ (\alpha_1, \alpha_2, \cdots, \alpha_k, g^{x_{i+1}}), K_{i+1}([(0, 0), \cdots, (BF_{dest}, g^{dest}), \cdots, (BF_{i+1})], sig, K_t)\rfloor$

其中, α_i 为计算对称密钥 K_{i+1} 的参数, 总共有 k 个, k 在正常应答时值为 1, 在自动应答时为大小固定的可配置参数, 表示接收一次自动路由应答的最大前驱节点个数, 值得一提的是增加 α_i 个数较重新发送分组的通信代价要小得多; K_{i+1} 是对称密钥, 节点 M_{i+1} 用其加密报文的最后三部分; BF_{i+1} 是节点 M_{i+1} 计算的 Bloom filter; K_t 是路由请求报文中的一次性私钥; sig 为路径上节点对路径链表的累计签名(Aggregate Signature), 即 $n+1$ 个节点各自对路径链表 $n+1$ 个不同部分的签名合成而得的短签名.

路径链表 $RList = [(0, 0), \cdots, (BF_{dest}, g^{dest}), \cdots, (BF_{i+1}, g^{x_{i+1}})]$ 的长度是固定值, 为 $Len = (HOP_{max} + 1) \times Len_{unit}$, Len_{unit} 为每个节点加入链表的数据单元长度, 即$(BF_{i+1}, g^{x_{i+1}})$的比特长度, Len_g 为 $g^{x_{i+1}}$ 的长度; 节点 M_{i+1} 在发送路由应答报文时, 将 $g^{x_{i+1}}$ 移到明文部分. (BF_{dest}, g^{dest}) 为目的节点加入 RList 的数据单元, $BF_{dest} = H2(K_{SD}(seq, S), g^{dest})$, 其长度与 BF_{i+1} 相同. $HMAP: \{0, 1\}^* \rightarrow F_q$, 为满足密码学要求的散列函数, 将任

意的二进制数散列到域 F_q 中; $HMAP': F_q \rightarrow \{0,1\}^*$ 将域 F_q 中元素散列为二进制数, 为满足密码学要求的散列函数. 节点 M_i 对接收到的路由应答报文处理过程如下:

(1) 令 $T_{i+1,i} = g^{x_i x_{i+1}}$, $x_i = HMAP(T_{i+1,i} \| RD_S)$, 计算:

$$K'_{i+1} = \alpha_L x_i^k + \alpha_{L-1} x_i^{k-1} \cdots + \alpha_1 x_i \quad (2)$$

$K_{i+1} = HMAP'(K'_{i+1})$, 解密 $K_{i+1}(\cdots)$; 再用 K_t 解密记录中的 $\{H_h, END\}_{PK_i}$, 解密所得数据若不正确则终止处理;

(2) 判断处理终止的条件: k 值大于 1 且 $g^{x_{i+1}}$ 存在于路由堆栈中; 报文接收时间不在时间窗内; 用 Bloom filter 检验到路径链表 RList 为重复发送; 否则继续处理;

(3) 若为第一次收到路由应答, 建立时间窗; 计算 g^{α_i}, 为本节点与源节点共享的秘密数; 若 $g^{x_{i+1}}$ 在路由堆栈中不存在, 则将 $g_{next} = g^{x_{i+1}}$ 压入堆栈;

(4) 对路由表中所有 g^{α_j} (不包括 g^{α_i}), 计算 $H(g^{\alpha_j} \| RD_s \| g^{x_i})$, 将计算所得结果用 2.4 节的方法编码为向量 BF_i, 由此得到本节点在路径链表中的数据单元 (BF_i, g^{x_i});

(5) 对更新过的路径链表 RList = $[(0,0), \cdots, (BF_{dest}, g^{dest}), \cdots, (BF_{i+1}, g^{x_{i+1}}), (BF_i, g^{x_1})]$ 最右 $2Len_{unit}$ 位数据计算签名, 将其加入到累计签名 sig 中;

(6) 计算 $IM_i = g^{\alpha_i} \| RD_s$, $IM = H(IM_i \| 1) \| H(H(IM_i \| 1) \| 2) \| \cdots$, $Sizeof(IM) = Len$, $IM = IM \odot (ZERO)_{len_g}$, 将 IM 最右 Len_g 位置零,

$$RList = RList \oplus IM \quad (3)$$

处理的目的是盲化路径链表, 使节点无法读取路由与推断到目的节点的距离;

(7) 从路由堆栈中选择一个未打处理标记的 g_{last}, 并打上处理标记, 若无这样的 g_{last}, 从所有的 g_{last} 中选择一个, 将选出的 g_{last} 组成集合 B;

(8) 若为时间窗监控机制在条件满足时启动的自动应答过程, 则从路由堆栈中选出不多于 k 个未打标记的 g_{last} 组成集合 B, 并对其打上处理标记; 另外其处理也经过(4)~(6)的步骤; 如果还有 g_{last} 未被标记, 可以在应答发送后递归处理;

(9) 对 B 中每个元素 $g_{last} = g^{x_{i-1}}$ 计算: $T_{i,i-1} = g^{x_i x_{i-1}}$, $HMAP(T_{i,i-1} \| RD_S)$;

(10) 随机选择 $K'_i \in F_q$, 计算 $K_i = HMAP'(K'_i)$ 作为对称密钥, 解以 $(x_{i,i-1}, K'_i)$ 为已知坐标对、以 α_j 为待定元由方程(2)组成的 k 元一次方程组(或一元一次方程), 解出 α_j;

(11) 将 RList 中的 g^{x_i} 移到明文部分, 使用 K_i 加密修改后的 RList、sig 与 K_t, 结合步骤(10)中得到的结果, 形成新的路由应答报文; 并向外发送新的路由应答报文.

当路由应答报文到达源节点时, 若源节点已处理的路由应答超过设定的限值, 则直接丢弃此路由应答报文. 源节点首先同中间节点一样检查路由应答报文的合法性. 其次, 源节点处理路径链表并抽取路由; 获取 g^{x_1} 后, 通过 g^{x_1} 可以计算出 g^{α_1}, 用公式(3)计算得到 RList 中的 BF_1 与 g^{x_2}; 将 RList 右移 Len_{unit} 位用同样的方法连续处理, 直到得到 (BF_{dest}, g^{dest}), 验证它是否为目的节点的数据单元, 如果不是则丢弃此路由应答报文. 最后验证累计签名的正确性, 如果不正确则终止处理, 由处理过程得到由共享秘密数所决定的一条路由 $g^{\alpha x_1} \rightarrow g^{\alpha x_2} \rightarrow \cdots D$. 随着路由应答报文的不断到达, 源与目的间的网络拓扑信息越来越丰富. 若通信的目的节点未变, 可以利用前次会话期间所得到的链路评估结果, 但需要解决前次获得的路由节点是否与现在获得的路由节点为同一个节点的问题, 此时可利用返回的 BF_i 来确定两个伪名是否对应同一个节点.

3.3 匿名数据传输

源节点用文献[9]中 B.3 与 B.4 节公式计算图上节点出边的概率分布, 逐跳选择一条到目的的路径, 假设为 $g^{\alpha x_1} \rightarrow g^{\alpha x_2} \rightarrow \cdots g^{\alpha x_n} \rightarrow D$. 在数据分组尾部添加确认报文的第二返回路径索引表 NRT_0 及其相关随机数 RD_0, 作为数据的一部分, 第二路径的计算用文献[14]中 B.5 节公式完成. 处理数据分组时, 如果数据长度达不到固定值, 填充随机数据到固定长度 Len_D. 传输过程中节点 M_i 接收到的报文格式如下:

$$\lfloor N, H_{K_{i,i-1}}(H(N \| C'^1_{i-1}), RD_i, NRT_{i-1}), C'_{i-1} \rfloor$$

N 是节点 M_{i-1} 产生的非减数值; C_{i-1} 是经过节点 M_{i-1} 盲化的密文数据; $K_{i,i-1}$ 是通过节点 M_{i-1} 与 M_i 共享的秘密数 $g^{x_i x_{i-1}}$ 导出的对称密钥; $H_{K_{i,i-1}}$ 是以 M_{i-1} 与 M_i 共享的对称密钥 $K_{i,i-1}$ 为参数的快速单向函数; RD_i 是节点 M_{i-1} 对随机数的解密; NRT_{i-1} 是节点 M_{i-1} 计算的下一跳节点索引表, NRT_{i-1} 与 NRT'_{i-1} 长度固定为 L, $L = Sizeof(g^{\alpha_i})$, $L_{index} = L/(HOP_{max} + 1)$; 报文头作用同文献[3].

在源节点上数据打包过程如下:

(1) 令 $RD_n = K_{s,n}(RD_0)$, RD_0 为源节点选择的随机数, 也看作 RD_{n+1}, $K_{s,n}$ 由 g^{α_n} 导出的对称密钥, 迭代计算 $SH_i = H(g^{\alpha_i} \| RD_{i+1} \| 1)$, $RD_{i-1} = K_{s,i}(RD_i)$;

(2) 计算 $Nindex_i = h(H(g^{ax_i} \| g^{x_{i+1}} \| RD_{i+1}))$, 长度为 L_{index}, $g^{x_{i+1}}$ 为下一跳节点的伪名, $NRT_0 = 000\cdots \| NIndex_n \| NIndex_{n-1} \| \cdots \| NIndex_1$;

(3) 计算 $NRT_0 = NRT_0 \oplus (SH_1 \langle\langle L_{index}) \oplus (SH_2 \langle\langle 2L_{index}) \oplus \cdots \oplus (SH_{n-1} \langle\langle (n-1)L_{index})$, 左移后低位补零, 后同; 恢复 $SH_1 = H(g^{ax_i} \| RD_{i+1} \| 1)$;

(4) 设确认报文第二返回路径为 $D \rightarrow g^{\alpha_m} \rightarrow \cdots \rightarrow g^{\alpha x_1} \rightarrow S$, 计算 $RD'_m = K_{s,m}(RD'_0)$、$RD'_{m-1} = K_{s,m-1}(RD'_m)$、$\cdots$、$RD'_1 = K_{s,1}(RD'_2)$, $SH'_i = H(g^{\alpha_i} \| RD'_{i+1} \| 1)$;

(5) $Nindex_i = h(H(g^{\alpha_i} \| g^{x_{i-1}} \| RD'_{i+1}))$, $NRT'_0 = 000\cdots \| NIndex_1 \| NIndex'_2 \| \cdots \| NIndex'_m$, $NRT'_0 = NRT'_0 \oplus (SH'_{m-1} \langle\langle L_{index}) \oplus (SH_{m-2} \langle\langle 2L_{index}) \oplus \cdots \oplus (SH'_1 \langle\langle (m-1)L_{index})$; 将 NRT'_0 与 RD'_0 加到数据分组的尾部.

(6) 计算 $SH_i = SH_i \| H(SH_i \| 2) \| H(H(SH_i \| 2) \| 3) \| \cdots$ 到长度 len_D;

(7) 用源与目的间密钥加密后的数据密文记为 C, 计算 $C'_0 = C \oplus SH_n \oplus SH_{n-1} \oplus \cdots \oplus SH_1$, C'_0 为盲化的秘文;

(8) 选择 N, 计算 $H(N \| C^1_0) H_{K_{s,1}}(H(N \| C^1_0), RD_0, NRT_0)$;

在中间节点 M_i 上的处理过程如下:

(1) 解密 $H_{K_{i,j-1}}(\cdots)$, 得到 $H(N \| C^1_{i-1}), RD_{i-1}, NRT_{i-1}$, 验证 $H_i(N \| C^1_{i-1})$ 是否合法, 若不合法, 或用 Bloom filter 检验

出 NRT_{i-1}、RD_i 及 $H(C'_{i-1})$ 重复, 则终止处理;

(2) 用 $K_{s,i}$ 解密 RD_i 得 RD_{i+1}; 计算 $NIndex_i = Low_{L_{index}}(NRT_{i-1})$, 即取 NRT_{i-1} 最低 L_{index} 位值; 选取长度为 L_{index} 的随机数 RD, 令 $SH_i = H(g^{ax_i} \| RD_{i+1} \| 1)$, 计算 $NRT_i = (((NRT_{i-1} \oplus RD) \oplus (SH_i \ll L_{index})) \gg\gg L_{index})$, 算后恢复 SH_i 值;

(3) 逐个读取本路由表项的路由堆栈中所存储的伪名 g' (不包括 $g^{x_{i-1}}$), 计算 $h(H(g^{ax_i} \| g' \| RD_{i+1}))$, 看是否有与 $NIndex_i$ 相等者, 相等则下一跳节点为 $g^{x_{i+1}} = g'$, 并算得相应的 $K_{i,i+1}$, 如果都不相等则终止处理;

(4) 计算 $SH_i = SH_i \| H(SH_i \| 2) \| H(H(SH_i \| 2) \| 3) \| \cdots$ 到长度 Len_D, $C'_i = (C'_{i-1} \oplus SH_i)$;

(5) 选择 N, 计算 $H(N \| C^1_{i-1})$, $H_{K_{i,j+1}}(H(N \| C^1_{i-1}), RD_{i+1}, NRT_i)$;

(6) 将重新计算的数据包发送到下一个节点.

3.4　数据分组的确认

目的结点收到源节点发来的数据分组后, 从两条路径返回确认报文, 一条沿主路径返回, 另一条按源节点选定的第二路径返回. 确认报文在节点 M_i 上数据格式如下:

$$\lfloor (N, H_{K_{i,i+1}}(N \oplus ACK, NRT_{i+1}, RD_{i+1}) \rfloor$$

其中, N 含义同上节; ACK 为一数值, 表示此报文为确认报文; RD_{i+1} 为随机数, 当确认报文沿主路径返回时, $RD_{i+1} = H(C'_i)$, $H(C'_i)$ 用来标识确认的是哪个数据分组, C'_i 含义同上节; NRT 是返回确认报文的节点索引, 长度固定为 L, 在主路径上初值为随机数.

主路径上返回确认报文时, 路径上每个节点都需要发送确认报文, 从目的节点开始反向合并成一个报文返回. 当存在断链或异常链路时, 路径被分为靠近源节点的与靠近目的节点的两部分, 靠近源节点的节点依然可以正常发回确认报文. 如果中间节点等待超时接收不到确认报文, 则自己创建确认报文, 沿途合并其他节点的确认形成一个统一的报文. 为了避免多个中间节点都创建确认报文, 形成多次确认, 超时值应该设为本节点路由请求与路由应答的时间差上限加上常数的富余量. 中间节点 M_i 上对 NRT_{i+1} 处理过程如下:

(1) 计算 $NIndex_i = h(H(g^{ax_i} \| g^{x_{i+1}} \| H(C'_i)))$, $g^{x_{i+1}}$ 为上一跳节点的伪名, 在创建确认报文的节点上为自身伪名, $H(C'_i)$ 是节点 M_i 匿名传输的盲化数据 C'_i 的一个散列, 是对确认报文的一个标识, 表示确认的是哪个报文;

(2) $NRT_i = (NRT_{i+1} \ll L_{index}) \oplus NIndex_i$, 相当于将 NRT_{i+1} 最低 L_{index} 改为 $NIndex_i$;

(3) $SH_i = H(g^{ax_i} \| H(C'_i) \| 1)$;

(4) $NRT_i = NRT_i \oplus (SH_i \ll L_{index})$;

在目的节点给确认报文选择第二条返回路径时, 它利用数据部分携带的 NRT'_0 与 RD^l_0 来选定路径. NRT_{i+1} 与 RD^l_{i+1} 在目的节点向外广播确认报文时分别等于 NRT'_0 与 RD'_0. 第二路径上的中间节点 M_i 上对确认报文处理过程如下:

(1) 若验证报文头不合法, 或用 Bloom filter 检验出 NRT_{i+1} 与 RD_{i+1} 重复, 则终止处理;

(2) $NIndex_i = Low_{L_{index}}(NRT_{i+1})$, $RD_i = K_{s,i}(RD_{i+1})$, $SH_i = H(g^{ax_i} \| RD_{i+1} \| 1)$;

(3) 逐个读取本路由表项的路由堆栈中所存储的伪名 g' (不包括 $g^{x_{i+1}}$), 计算 $h(H(g^{ax_i} \| g' \| RD_{i+1}))$, 看是否有与 $NIndex_i$ 相等者, 如有相等, 则下一跳节点为 $g^{x_{i-1}} = g'$ 并算得相应的 $K_{i,i-1}$; 若都不等于 $NIndex_i$, 则终止处理;

(4) 将 NRT_{i+1} 最低 L_{index} 位置为随机数, 再计算 $NRT_i = (NRT_{i+1} \gg\gg L_{index}) \oplus SH_i$.

(5) 选择 N, 计算 $H_{K_{i,i-1}}(N \oplus ACK, NRT_i, RD_i)$, 然后向外发送确认报文.

如果源节点最终收到表示第二条路径正确返回应答报文.

源节点收到确认报文后, 根据节点的应答情况用文献[9]中 B.2 节方法进行所需计算.

4　匿名与安全分析

4.1　匿名分析

(1) 身份机密性　在路由请求与路由应答报文中出现的所有源与目的节点真实身份都用源与目的共享的密钥加密, 其他节点无法获得. 源的伪名与真实身份没有联系, 无法从伪名获得有关身份的信息; 另外, 每次会话使用不同的伪名, 伪名之间是不可比较的, 无法从伪名之间的关系推断出源节点任何身份信息. 中间节点也使用伪名来隐藏身份; 虽然源节点未验证中间节点的真实身份, 还是可以通过对其路由行为的反馈监控, 来判断中间节点是否可靠.

(2) 位置机密性　文献[1]中路由请求报文长度泄漏了节点到源节点距离的信息; 对文献[2]中的路由请求与应答报文来说, 路由上的节点可通过其"洋葱"部分真实长度计算出到源节点的距离. 本文中报文长度为固定值, 窃听者无法从长度上推断出位置信息. 文中设计的路由应答与确认报文中没有步跳计数, 路由请求报文中的 U 值本身也不表示到源节点的距离. 通过监听不同节点发送的路由请求报文, 比较其 U 值也只能计算出到源节点的相对远近, 在不能监听处理全网数据流、也不掌握全网组成等信息的情况下, 无法用 U 值确定节点到源的距离.

(3) 路由的匿名性　攻击者做法之一是通过对报文内容进行比较或计算特征量来进行追踪. 本文的设计中, 除了路由请求广播报文外, 其他报文的内容经过逐跳的加解密或盲化处理, 通过窃听难以追踪路由. 通过侵蚀路径上的节点, 攻击者可获得被侵蚀节点的前后各一跳的路由信息; 若所侵蚀的节点是传输路径上相继的节点, 则获得一段路由信息, 并可追踪数据分组在此段路由上的传输; 若路径上节点都被侵蚀, 则数据分组可被全程追踪; 若侵蚀的不是路径上的节点, 则不能追踪报文. 为了进行定量分析, 用类似文献[2]的路由可追踪率概念来定义报文可追踪率: 假设传输路径上节点总数为 L, 被侵蚀的路段数为 K, 被侵蚀的第 i 路段有 F_i 个节点, 则报文可追踪率为 $\left(\sum_{i=1}^{K} F_i \times F_i \div L\right) \div L$. 在仿真试验中, 我们用此公式分析数据分组传输的可追踪性. 本设计中, 报文的可追踪性要优于文献[2,3]中的方法, 因为在文献[2,3]的设计中, 路径上任意两个被侵蚀节点可通过比较报文的数据部分异同

来追踪数据分组的传输.

攻击者做法之二是通过对报文收发的时间顺序进行分析来追踪报文. 为了抵御这种攻击, 若节点比较繁忙, 有多个报文分组需发送, 对同优先级的分组不完全采用先到先服务原则, 而是采取一种乱序的方法来发送; 对比较空闲的节点在发送缓冲中注入假报文分组.

4.2 安全分析

(1) 被动攻击 被动攻击通常是通过拒绝执行指定的路由功能完成的. 对这种攻击, 文献[1, 2, 3]都没有进行处理, 因为一是无法定位问题节点的位置, 二是重新发起路由发现过程可能依然选择这条问题路径. 本文采取数据分组确认机制进行反馈学习, 来降低对问题链路信任度, 减小其概率值, 然后选取其他路径来执行路由功能; 用同样的方法可以处理节点的拜占庭行为. 对于窃听和分析, 主要通过加密、盲化与注入伪数据包来防止.

(2) 拒绝服务攻击 拒绝服务攻击分为多对一的攻击与一对多的攻击. 对于多对一的攻击, 由于源节点与目的节点位置的机密性, 攻击者找不到目标进行攻击. 对于一对多攻击, 本文的两种方法可有助于防止攻击, 其一是在路由请求中的对目的节点检验通过对称密钥解密而减小节点计算量, 其二在路由应答与确认报文中, 采取每跳验证的方法检验报文的合法性与重复性.

(3) 虫洞攻击 虫洞攻击是攻击者在一个节点获取报文分组后, 用隧道的方式传到另一个节点, 不破坏原来报文中的任何部分, 它难以被检查. 虫洞形成后, 攻击者可进行选择性通过之类的拜占庭行为, 也可与其他方法结合分割控制网络.

在本文的设计中, 通过对数据分组的确认来检验节点行为状况; 以此反馈机制防止虫洞的拜占庭攻击, 相当于把虫洞当作一个单跳链路, 不是阻止它的形成, 而是避免虫洞的拜占庭行为所带来的影响. 由于通信的匿名, 攻击者很难使用流量分析的方式发现源与目的之间的关键节点, 从而防止攻击者控制关键点形成虫洞与分割控制网络.

(4) 报文篡改与重放攻击 对路由请求报文而言. 被侵蚀节点对 U 值进行于己有利的修改是将接收到的 U 值不做修改广播出去, 从而形成经过被侵蚀节点的路径; 但这在信道公平分配的情况下它的影响是局部的, 随后的反馈学习过程可检验被侵蚀节点的异常行为.

对路由应答报文而言. 篡改路径链表会被源节点检验出来, 其对中间节点的资源消耗只能存在一段时间, 且在公平信道上无法阻止其他正常路由应答报文的传送. 路由应答路径上连续两个被侵蚀节点可以合谋使用不同的伪名来伪造多个路由应答报文, 同上节 3) 所述一样, 我们不是防止它的形成, 而是避免它们异常行为对路由性能的影响; 另外由于时间窗的限制, 敌手也只能伪造有限个路由应答, 同样它也无法阻止正常路由应答报文的传送.

对数据传输与数据确认而言, 中间节点通过使用 Bloom filter 检验路由节点索引表、随机数及报文散列的重复性来阻止被侵蚀节点的重放攻击. 修改路由节点索引表及与之相应的随机数则不能进行路由, 等同于被动攻击.

5 性能试验与分析

5.1 模拟环境

在我们的加密实现中, 为了节省带宽, 涉及公钥的部分使用椭圆曲线密码体制. 对称密钥的长度为 128 比特, 伪名的长度为 160 比特, 源节点与目的节点真实身份为 32 比特, 源节点公钥长度为 320 比特, 随机数的长度为 160 比特, *seq* 序列号长度为 128 位. 路由应答报文中各个路由中间节点加入的数据单元(BF_i, g^{x_i})长度为 260 比特, 其中, g^{x_i}为 160 比特, BF_i为 100 比特, 100 比特的 BF_i 在采用 8 个独立的散列函数、中间节点同时为 8 个不同的源与目的的会话充当路由节点时错判的概率接近千分之二, 是一个比较合理的均衡数值. 整个路由的中间节点最大步跳数设为十, 在路由应答中整个路径链表的长度为 2860 比特, 350 多个字节; 自动路由应答进程向前驱节点组播时, 计算对称密钥的参数个数 k 选 3 是一个较合适的值, 则参数总长度为 384 比特; 累计签名长度为 160 比特. 确认报文中的 NRT 与 NRT' 长度为 160 比特, RD' 长度为 160 比特. 在模拟中, 作一次公钥加密操作的时延为 150ms 左右, 解密为 38ms 左右, 对称密钥加密速度为 32.3Mbps, 对称密钥解密速度为 32.1Mbps.

我们使用 GloMoSim 作为模拟的平台, 模拟 50 个节点放入 1000m × 1000m 的区域中, 初始随机布置. 每个节点的传输范围是 250m, 信道传输速率位 2Mb/s. 节点运动模型采用节点随机移动模型, 节点以小于 10m/s 的任意速度与方向, 运动到任意的位置, 在每个位置停留小于 300 秒的时间(采用 0, 30, 60, 120, 300s 试验). 网络中设定十二个并发数据会话, 以 CBR 方式产生网络数据流, 会话的源与目的随机选择与变化, 每秒产生 4 个数据分组, 每个分组有效载荷大小为 512 个字节. 使用 IEEE802.11 的 DCF 作为媒体访问控制协议. 每次仿真运行 300 秒, 结果是多次运行的平均值.

5.2 结果及分析

我们实现文献[8]中的 SMR(split multipath routing) 作为比较对象, 使用其在两条路径都失效情况下重新发起路由发现过程的方法.

从图 2 可以看出, SMR 协议的平均端到端的传输时延要小于本文设计的路由协议的时延值, 这是由于: 其一是本协议中通信的目的端点返回更多的报文, 引起碰撞造成的; 其二是在本协议中, 中间节点处理加解密时延加长. 但由试验结果来看, 所增加的时延很小, 增加匿名安全性而带来的代价是可以接受的. 当节点移动性增加时, 时延增长, 这是由于高移动性导致路由变化, 包丢失率增加, 超时机制应用增多的效果.

从图 3 可以看出, 本协议传输成功率与 SMR 相比有较大的提高, 这是因为本协议拥有更充分更有效的网络状态, 反馈

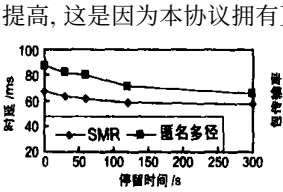

图 2 端到端平均时延

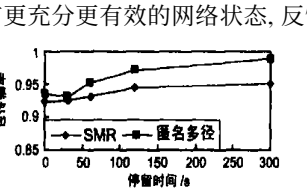

图 3 传输率比较图

学习,使源节点及时选择比较稳定的路径传输数据包,当移动性降低时本协议传输成功率接近于 1.

图 4 表示增加混淆伪数据包时,中间节点上平均增加的负荷,用节点在仿真期间混淆包数与正常数据包数比率表示,横轴为混淆而开的数据包缓冲区大小,不同的曲线为混淆时间窗大小. 从图上可以看出,时间窗增大,所产生的混淆伪数据包减少,一般情况下,伪数据包产生率小于 1.

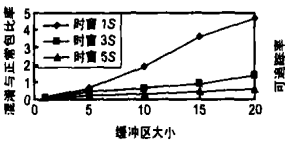

图 4　混淆负荷

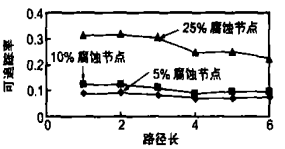

图 5　可追踪率比较

从图 5 可以看出,随着路径的加长,可追踪率下降;当网络中被侵蚀的节点数目增多时,可追踪率上升. 由此可以看出匿名是相对的,随着敌手攻击能力的增强,包括侵蚀节点能力提高与监听范围扩大,对数据分组的追踪能力快速提高,协议的匿名性降低. 但是在实际环境中敌手的能力也要受到各种条件限制.

6　结论及今后的工作

路由的安全匿名是战术无线自组网等类似系统整体安全解决方案中非常重要的一部分. 在本文中,首先给出了匿名的定义,然后利用单私钥多公钥的密码体制、Bloom Filter 方法、轻型洋葱盲化算法、强化学习算法,在改进的移动自组网络多径路由发现的基础上,设计了移动自组网络安全匿名的多径路由协议. 对协议的安全性与匿名性进行了详细分析,并对性能进行了仿真,通过仿真与分析得知:所设计的协议在基本不牺牲路由性能的基础上,提供了较好的可靠性与可用性,并且达到了所定义的匿名性,能够抵御多种攻击,增强了系统的安全性. 今后工作的一个重要方向是在已有的路由发现基础上,设计一种次路由发现过程,主要利用路由发现主过程发布的信息,以较小的代价探测路由变化与未发现的路由,对已有的路由信息进行部分更新;另一个重要方向是,通过在不破坏安全性与匿名性的情况下提供链路失效后的本地修复功能,提高系统效率.

参考文献:

[1] K El-Khatib, L Korba, R Song, G Yee. Secure dynamic distributed routing algorithm for ad hoc wireless networks[A]. In International Conference on Parallel Processing Workshops (ICPPW'03)[C]. Kaohsiung, Taiwan: IEEE Computer Society, 2003. 359–366.

[2] J Kong, X Hong. ANODR: ANonymous on demand routing with untraceable routes for mobile ad-hoc networks[A]. In Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'03)[C]. Annapolis, Md, USA: ACM, 2003. 291–302.

[3] Bo Zhu, Zhiguo Wan, et al. Anonymous Secure Routing in Mobile Ad-hoc Networks[A]. the 29th Annual IEEE international Conference on Local Computer Networks[C]. Tampa, USA: IEEE Computer Society, 2004. 102–108.

[4] Y C Hu, D B Johnson, A Perrig. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks[A]. In Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002)[C]. NY, USA: IEEE Computer Society, 2002. 3–13.

[5] Y-C Hu, A Perrig, D B Johnson. Ariadne: A secure ondemand routing protocol for ad hoc networks[A]. In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002)[C]. Georgia, USA: ACM, 2002. 12–23.

[6] A Nasipuri, S R Das. On-Demand Multipath Routing for Mobile Ad Hoc Networks[A]. Proceedings of IEEE ICCCN'99[C]. Boston, USA: IEEE Computer Society, 1999. 64–70.

[7] J Raju, J J Garcia-Lunar Aceves. A New Approach to On-demand Loop-Free Multipath Routing[A]. Proceedings of IEEE ICCCN'99[C]. Boston, USA: IEEE Computer Society, 1999. 522–527.

[8] Sung-Ju Lee, Mario Gerla. Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks[A]. In Proceedings of the IEEE ICC, 2001[C]. Helsinki: IEEE Computer Society, 2001. 3201–3205.

[9] Baruch Awerbuch David Holmer Herbert Rubens. Provably Secure Competitive Routing against Proactive Byzantine Adversaries via Reinforcement Learning[R]. Technical Report Version 1, May 16, 2003. Computer Science Department, Johns Hopkins University.

[10] Nicolao Cesa-Bianchi, Yoav Freund, David P Helmbold, David Haussler, Robert E Schapire, Manfred K Warmuth. How to use expert advice[J]. Journal of the Association for Computing Machinery, 1993. 382–391.

[11] Brent R, Waters Edward W, Felten Amit Sahai. Receiver Anonymity via Incomparable Public Keys[A]. Proceedings of the 10th ACM conference on Computer and communications security[C]. Washington D C, USA: ACM, 2003, 112–121.

[12] B Bloom. Space/time Tradeoffs in Hash Coding with Allowable Errors[J]. Communication of ACM, 1970, 13(7): 422–426.

[13] 郭晓峰, 陈跃泉, 陈贵海. 一种累计多路径的移动自组网络路由策略[J]. 软件学报, 2004, 15(04): 594–603.

作者简介:

章 洋 男, 1970 年 10 出生于安徽安庆, 现为中国科学院软件研究所博士生, 大学本科就读于中国矿业大学, 2001 年进入软件所, 主要研究领域为网络安全与分布式计算.

E-mail: Zhang_yang_own@yahoo.com.cn.

范植华 男, 1942 年出生, 现为中国科学院软件所研究员, 博士生导师, 目前主要研究方向为: 并行编译、集群计算等.

E-mail: Fan_Zhihua@hotmail.com.

SPW/Prosim 2006 国际研讨会征文通知

会议主题: 软件过程变革——应对挑战

信息技术的发展, 迫使软件的开发方法和质量管理面临新的挑战。快速、高质量地交付软件产品、有效地进行风险控制一直是软件工程追求的目标。新的软件过程技术以及软件复用、软件演化等技术日益发展并受到重视。2006 年, 两个成功的国际研讨会 SPW, ProSim 将首次携手共同探讨这些问题。会议将于 2006 年 5 月 20~ 21 日, 在中国上海, 与 ICSE2006 同时召开。

会议的主题是"软件过程变更——应对挑战" (software process changer meeting the challenge). 内容包括: 世界著名软件过程领域研究者的特邀报告、针对软件过程挑战与解决方法的论文报告、工具演示、关于软件过程研究方向的专题讨论会。

一、征文范围(包括但不限于)

欢迎有关软件过程的经验、描述和方法等各相关研究领域的论文。例如, 过程内容(文档驱动的、变化驱动的、体系结构驱动的、风险驱动的、涉众驱动的……); 过程表示与分析; 过程工具和度量; 过程中的人为因素; 过程建模; 过程模拟等等。会议将出版正式论文集, 优秀论文还将推荐到国际重要学术刊物 International Journal of Software Process: Improvement and Practice.

二、征文要求

1. 论文未被其他会议、期刊录用或发表;
2. 论文需用英文书写, 长度为 10 页或 10 页以内;
3. 来稿采用本会议电子投稿系统, 格式为 PDF 或 MS Word;
4. 详见会议主页

http: www.cnsqa.com/~spwprosim2006;

http://www.iscas.ac.cn/~spwprosim2006;

三、重要日期

论文提交: 2006 年 1 月 6 日

录用通知: 2006 年 2 月 17 日

最终论文: 2006 年 3 月 12 日

四、联系方式

联系人: 舒风笛

电话: +86-10-62612440 传真: +86-10-62550138

电子邮件: spwprosim2006@iscas.ac.cn

地址: 北京中关村南四街 4 号, 中国科学院软件研究所