Ad Hoc 网络的数据安全传输方案研究*)

卢社阶　崔国华　陈　晶　刘志远

(华中科技大学计算机科学与技术学院　武汉 430074)

摘　要　由于 Ad Hoc 网络结构的特点,使其更容易遭到攻击,安全的路由和安全数据传输已成为 Ad Hoc 网络研究的热点。对于网络的外部攻击,在不同的假设条件下,已经提出了一些有效的抵制方案。但对于内部攻击,还没有一种快速、准确、可行的对恶意行为进行预防和检测的方法,一般采用信誉机制加以解决,但存在较多缺点。本文基于 Reed-Solomon 编码的纠错技术,提出了一种在存在 Byzantine 攻击节点的网络环境下的安全数据传输协议(Secure Data Transmission Protocol,SDTP)。该协议不仅能实现数据的安全传输,而且能准确判定恶意攻击的发生和攻击节点的位置,为在 Ad Hoc 网络中检测内部攻击节点提供了一种更准确、具体且实际可行的算法,该算法也可被用于安全路由协议中。

关键词　Ad Hoc 网络,安全数据传输,入侵检测,恶意行为判别

Research on Secure Data Transmission for Ad Hoc Networks

LU She-jie　CUI Guo-hua　CHEN jing　LIU Zhi-yuan

(College of Computer Science & Technology, Huazhong University of Science & Technology, Wuhan 430074, China)

Abstract　Ad Hoc networks are vulnerable to some kinds of attacks for its network structure characteristics, this makes the secure routing and data transmission be research hotspots of Ad Hoc applications. As to the External attacks, some efficient and secure schemes have been propose to resist them, but to the internal attacks, there is still no practical and efficient way to guard against and detect the malicious behavior. To resolve the problem, reputation mechanism is usually used, but it is not a favorable way. Based on the Reed-Solomon error-correct coding scheme, we propose a secure data transmission protocol (SDTP) to obtain a secure data transmission in a Byzantine attack existing environment. Besides the secure data transmission, it can judge the malicious behavior and transmission error. More importantly, it also can locate the malicious node accurately. The algorithms of the protocol can be used in secure routing protocol.

Keywords　Ad Hoc network, Secure data transmission, Intrusion detection, Malicious behavior judgment

1　引言

无线 Ad Hoc 网络是随着无线通信技术的快速发展而出现的一种新型网络,是一组带有无线收发装置的移动节点组成的一个多跳的临时性自治系统,具有机动性、快捷性等特点,广泛应用于军事战术通信、应急通信、协同移动通信、无线接入系统和传感网络等众多领域。Ad Hoc 网络中信息通信可分为两个阶段:路由发现阶段和数据传输阶段。安全路由协议(Secure Routing Protocols,SRPs)是安全信息通信的基础,这部分的研究比较活跃,相继提出了一些安全路由协议或可用的安全机制。如基于 DSR 协议,采用广播认证模式 TESLA,提出了 Ariadne[1];基于 DSDV 协议,利用单向 Hash 链认证技术,提出了 SEAD 协议[2];利用非对称密钥技术,提出了 ARAN[3];采用数字签名提供的端到端认证技术实现了 AODV 和 DSR 的协议安全机制[4]等。这些协议或机制大多只考虑了抵御外部攻击节点的情况,对于内部攻击,只能部分地解决或通过一些假设加以回避。

除了路由攻击以外,攻击者还可能攻击分组转发操作。分组转发操作攻击并不破坏每个节点的路由协议和路由状态,而是故意采用与路由状态相矛盾的方式交付数据分组。例如,攻击者可能沿着已建立的路由将数据分组丢掉、修改数据分组的内容,或者复制已经转发过的数据分组。这就是典型的 Byzantine 攻击。

路由层中针对内部攻击所采取的措施在数据传输层是可供借鉴的。如 Cheung 等提出了一种通过可信节点向其邻节点发送探测包的方法[5],实现联合入侵检测,但可信节点的界定是很困难的;Goodrich 提出一种"leap-frog"模式[6],能在两跳内不超过一条问题路由的前提下发现恶意节点等。这些模式在 Ad Hoc 网络的实际应用中都有很大的局限性。由于没有一种有效的恶意节点检测算法,文献[7,8]提出了在网络中建立节点信誉机制的方法,通过统计的方法逐渐明晰并限制恶意节点的活动。但这种机制在全网实现时开销很大,对恶意节点的活动反应很慢,且不准确,存在误判问题,不能有效区分恶意行为和网络传输错误。

在数据传输层针对内部攻击,文献[9]给出了一种基于 ACK 的检测方法,目的节点对成功接收到的每个分组回送一个 ACK 给源节点。如果在一条路由上丢失的分组数量大于可接受的门限,那么这条路由值得怀疑,源节点可以针对受到怀疑的路由初始化故障检测进程。源节点在其与目的节点之间进行二分搜索时,发送携带有中间节点列表的数据分组(称为探测器),探测器应该回送确认,源节点与每个探测器共享一个密钥,探测列表是加密的"洋葱"。每个探测器收到分组后,生成并采用共享密钥加密一个 ACK,然后将 ACK 回送给源节点,源节点再对加密 ACK 进行验证,将故障归于离目的

*)国家自然科学基金(No. 60403027)。**卢社阶**　博士研究生,主要研究领域为无线网络安全与路由协议安全与仿真;**崔国华**　教授,博士生导师,主要研究领域为信息安全、网络安全和密码学;**陈　晶**　博士研究生,主要研究领域为无线网络安全、路由协议安全及仿真;**刘志远**　博士研究生,主要研究领域为无线网络安全。

节点最近且回送 ACK 的那个节点。但这种方法的不足是十分明显的。首先，探测器的 ACK 要求通过一个安全信道回送到源节点，如何确定一条路由是安全可靠的本身就是个难题，并且可能从探测器到源节点间根本就不存在第二条路由，安全路由更无从谈起。其次，要求源节点已知其到达目的节点的每条完整路径，这对下层的路由协议提出了要求。在很多路由协议中，源节点可能并不知道到达目的节点的完整路由。此外，这种方式发现问题慢，且无法确认某种问题行为是否出于恶意，源节点的层层脱密运算量很大。

安全的路由协议能够将外部攻击节点排除在路由之外，这是安全数据传输的基本前提。但由于内部攻击节点的存在，且没有一种有效的抵制方法，或者即使路由协议能完全抵制内部攻击，这些内部攻击节点可能在路由发现阶段并不表现出来，因此也并不能保证数据传输阶段的安全。在无外部攻击节点存在的条件下，研究数据的安全传输是十分必要的。基于 Reed-Solomon 编码的纠错技术，本文提出了一种在存在 Byzantine 攻击的网络环境下的安全数据传输协议 SDTP，将要发送的消息数据经过 Reed-Solomon 编码后分别由多个不同路径传送，在接收端进行组装，在解码过程中可以检验是否存在篡改数据的恶意行为发生。若存在，则发起检测过程。该协议不仅能实现数据的安全传输，而且能准确判定恶意攻击的发生和攻击节点的位置。

本文后面的结构组织如下：第 2 部分介绍 Reed-Solomon 编码的特点及算法。第 3 部分描述 SDTP 协议。第 4 部分阐述恶意节点检测的方法。第 5 部分说明本协议所能抵制的内部攻击。第 6 部分给出本协议在多径路由协议 AOMDV 上运行的实验结果。最后给出本文的结论和下一步的工作。

2 Reed-Solomon 编码算法

Reed-Solomon 编码是一种基于有限域理论的非二进制形式的循环码[10,11]，是多元 BCH 码的一种，在现代数据通信和媒体中应用极其广泛。在发送端发送信息之前，编码器根据要发送的数据信息计算相应的纠错信息，并把纠错信息作为冗余校验和数据信息一起组成纠错码字。在接收端收到这些码字后，通过纠错解码器不仅能自动地发现错误，而且能自动地纠正码字在传输过程中的错误。

Reed-Solomon 编码可定义为：设 α 是伽罗瓦域的一个生成元，对于任一正整数 $t\leqslant 2^q-1$，存在一个能纠正 t 个符号(symbol)错误的 Reed-Solomon 编码，符号域为 $GF(2^q)$，有如下参数：

$$n=2^q-1, n-k=2t$$

式中，n 为码块长度，t 为能够纠正的错误数目，$n-k=2t$ 是校验码的符号数。对于一个信息码符多项式 $m(x)$，校验码生成多项式的一般形式为

$$g(x)=\prod_{i=0}^{k-1}(x-\alpha^{k_0+i})$$

通常取 $k_0=0$ 或 $k_0=1$，$n-k\geqslant 2t$。若取 $k_0=0$，则有

$$g(x)=(x+\alpha)(x+\alpha^2)\cdots(x+\alpha^{2t})=g_0+g_1x+g_2x^2+\cdots+g_{2t-1}x^{2t-1}+x^{2t}$$

式中 $g_i\in GF(2^q)$，$\alpha,\alpha^2,\cdots,\alpha^{2t}$ 是 $g(x)$ 的根。

$GF(2^q)$ 中的每个元素可以唯一地表示一个二进制 q 元组，即一个符号，因此可以用一个符号域为 $GF(2^q)$、参数为 (n,k) 的 Reed-Solomon 码来对二进制数据进行编码。一个包含 kq 比特的消息首先被分为 k 个 q 比特字节，每个 q 比特字节是 $GF(2^q)$ 中的一个符号，再将 k 个 q 比特符号按 Reed-Solomon 编码规则编码成包含 n 个符号的码字。

2.1 Reed-Solomon 编码的编码算法

(1)计算消息多项式：$m(x)=m_0+m_1x+\cdots+m_{k-1}x^{k-1}$，其中 $m_i\in GF(2^q)$，$k=n-2t$。

(2)将 $x^{2t}m(x)$ 除以 $g(x)$，可以得到

$$x^{2t}m(x)=a(x)g(x)+b(x)$$

其中 $b(x)=x^{2t}m(x) \bmod g(x)=b_0+b_1x+\cdots+b_{2t-1}x^{2t-1}$，$b(x)$ 叫做校验多项式(parity check polynomial)，则 $c(x)=b(x)+x^{2t}m(x)$ 就是消息 $m(x)$ 的码字多项式。

2.2 Reed-Solomon 编码的解码算法

(1)计算：

$$c(x)=c_0+c_1x+\cdots+c_{n-1}x^{n-1}$$

$$r(x)=r_0+r_1x+\cdots+r_{n-1}x^{n-1}$$

$$e(x)=e_0+e_1x+\cdots+e_{n-1}x^{n-1}$$

其中：$c_i,r_i,e_i\in GF(2^q)$，$e(x)=r(x)-c(x)$ 是错误多项式，$e_i=r_i-c_i$ 是有限域 $GF(2^q)$ 中的一个符号。

(2)若 $e(x)$ 中包含 v 个错误，则 $e(x)=e_{j_1}x^{j_1}+e_{j_2}x^{j_2}+\cdots+e_{j_v}x^{j_v}$，错误点的数目为 $X_{j_1}=\alpha^{j_1}$，$X_{j_2}=\alpha^{j_2}$，$\cdots$，$X_{j_v}=\alpha^{j_v}$，然后利用 Froney 的算法，可求出错误值为 $Y_l=e_{j_l}$，$l=1,2,\cdots v$。

(3)恢复原消息码字多项式 $c(x)$，有 $c(x)=r(x)-e(x)$。

3 安全的数据传输协议

保护路由消息交换只是移动 Ad Hoc 网络层安全解决方案的一个组成部分，有可能一个恶意节点能够正确参与路由寻找过程，但不能正确转发数据分组，攻击者可以将自己收到的所有分组丢掉或篡改以后再转发。这种攻击与网络传输错误很难区分，对于问题节点准确位置的判断，也是最难于用确定算法来实现检测的。而目前用得较多的信誉机制，回避了对及时性和准确性的要求，求得一种与现实生活所用的合理判断，其缺点是显而易见的，只能说是一种寻求安全的无奈的选择。下面是本文所提出的一种健壮且安全的数据传输协议 SDTP 的具体操作过程，用以实现在存在恶意攻击的环境下尽可能准确传递数据，并追查恶意节点的位置。

3.1 前提假设及符号定义

3.1.1 前提假设

达成上述目的，需要有一定的前提条件，这些条件越少越宽松越与实际应用相符合，方案才越有实用价值。本文所提安全传输方案基于了以下假设：

(1)在网络路由层可以抵制外部攻击，使外部节点无法加入到路由。这一点对于现有的大多数安全路由协议来说是不难做到的。

(2)源节点和目的节点是可信的，且在源节点和目的节点之间至少存在一条不包含恶意节点的路由。这一假设是实际的，若两节点间不存在一条安全的路由，则在这两个节点间实现安全通信是困难的。在本协议中，即使不存在这样一条安全路由，数据也可以做到安全传输，只是无法检测恶意节点的位置。

(3)一条路由中只有不超过一个恶意节点。这一假设并不影响数据的安全传输，只是在恶意节点定位时被用到。

(4)每个节点有自己的私钥，其它节点持有该节点的公

钥。任意两个节点间存在一对共享密钥，这可以在网络初始化时实现或通过网络密钥管理模式来实现。这些共享密钥对可以通过密钥交换协议如 Diffie-Hellman[12] 来实现，在协议实现过程中应保证密钥的安全性。

3.1.2 符号定义

为了描述方便，对本文所用到的一些符号定义如表 1。

表 1 符号定义表

P	待传输的数据包	CP	包 P 经加密后的 Reed-Solomon 编码
CP_i	第 i 个信息传送片断	CP_i^A	节点 A 收到的 CP_i 片断
K_A^{-1}	节点 A 的私钥	KXY	节点 X 和节点 Y 的共享密钥
$H\langle M\rangle$	对消息 M 的散列值	$\langle M\rangle K$	用密钥 K 对消息 M 加密

3.2 数据发送端

源节点 S 要发送一个数据包 P 至目的节点 D，S 和 D 间可能存在多条路由。如在图 1 所示的网络拓扑结构中，S 到 D 的可达路径有：S—A—B—C—E—D，S—A—B—F—M—G—L—D，S—H—F—M—G—L—D，S—H—J—K—D，S—I—J—K—D 等，将 P 用 K_{SD} 加密，记 $EP=(P)_{K_{SD}}$，设 EP 的长度为 kq 比特（可用简单的填充方法保证这一点），设编码能够纠错的符号数为 t，则码字长度 $n=k+2t$，在 $GF(2^q)$ 求得 EP 对应参数为 (n,k) 的 Reed-Solomon 编码 CP，CP 长度为 nq 比特，编码的膨胀率为 $\frac{n}{k}$，可通过适当选取 t 值使膨胀率在应用中达到合理，再将 CP 采用以下步骤传送到目的节点 D。

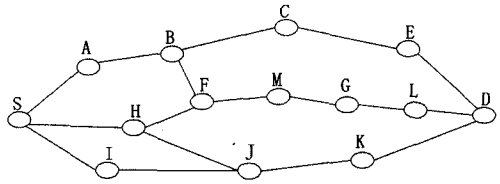

图 1 源节点 S 和目的节点 D 间的网络拓扑图

(1) 将 CP 分成 d ($2t<d\leqslant k$) 个片断 $CP_0, CP_1, \cdots, CP_{d-1}$，片断间可以不等长，任意片断 CP_i 所包含的比特数应可以被 q 整除，且长度不超过 tq 比特。值得说明的是，d 越大，传输过程能抵抗攻击的能力越强，但网络开销会随之有所增加。

(2)求出各片断 CP_i 的单向散列函数值 $H(CP_i)$，构建数据片断报文，格式为：

$[DATASEG, SOURCE, DEST, PK\_SEQ, SEG\_SEQ, CP_i, H(CP_i)]$

其中，

$DATASEG$：数据片断报文标识。

$SOURCE, DEST$：源节点和目的节点。

PK_SEQ, SEG_SEQ：数据分组序号和分组片断序号。

将同一数据分组的不同分组片断报文按不同的路径发送 D，不同的路径可以包含相同的节点，即可以是相交的多径路由。

3.3 中间节点

中间节点收到一个片断后，首先取出 $H(CP_i)$，看是否能在缓存中找到该值。若有，则可能是重放攻击，丢弃该包；否则，缓存 $H(CP_i)$。根据从包中取出的 CP_i 计算 $H(CP_i)$，比较该值与从包中取出的 $H(CP_i)$ 是否相同。若不同，说明数据传输错误，向 S 发送 ACK 报文，请求源节点重发该分组片断；若相同，转发该分组片断报文。

3.4 数据接收端

目的节点收到所有片断后，重组原始分组 CP，包含三个步骤：(1)计算校正因子(syndrome)；(2)计算错误位置；(3)计算错误值。当恶意节点修改校验码 b，则它也需要修改与之对应的码字 m_i。由于 $g(x)$ 的根也是 $c(x)$ 的根，接收端接收的码字 $r(x)=c(x)+e(x)$，$e(x)=\sum_{i=0}^{n-1} e_i x^i$ 是错误多项式。若所收到的码字完全正确，则 $r(x)$ 的根对应的 $g(x)$ 的函数值为 0。否则，任何错误将导致其值不为 0。校正因子计算公式为

$$S_i=r(x)\big|_{x=\alpha^i}=r(\alpha^i),\ i=1,2,\cdots,2t$$

如果存在 v 个错误，这些错误所在位置记为 $j_1, j_2, \cdots, j_v$，则 $e(x)=e_{j_1}x^{j_1}+e_{j_2}x^{j_2}\cdots+e_{j_v}x^{j_v}$，若记错误位置对应的错误值为 $Y_l=e_{j_l}$，$l=1,2,\cdots,v$，则可利用 Forney 算法求得错误值 Y_l，这样就求得了错误多项式：

$$e(x)=\sum_{l=1}^{v} Y_l x^{j_l}$$

进而恢复出 $c(x)$，$c(x)=r(x)-e(x)$，依此可求出消息多项式 $m(x)$，即多个分片的对应的分组的密文。最后目的节点用 K_{SD} 对密文解密即可得到源节点所传送的消息。

在上述过程中，若对编码进行解码时未发现错误，则 D 向 S 发 ACK，通知其分组已正确接收。如出错位置数超过编码的纠错能力，则请求源节点重发；若发现错误，但能纠正，则告知源节点，启动恶意节点位置检测过程。

4 恶意节点检测

Byzantine 攻击主要有三种表现，即窃听、丢包、篡改。对于窃听攻击，在协议中已经通过将数据 Reed-Solomon 编码片断用 K_{SD} 加密进行预防。并且，由于分组的所有编码片断并不经过完全相同的路由，一个恶意节点不能得到分组的所有编码片断，是无法重构分组加密信息的。对于丢包，可采用文献[13]中的方法加以检测。对于恶意篡改攻击，下面首先给出了恶意攻击和数据传输错误的判别，然后对恶意攻击节点的检测给出具体的方法。

4.1 恶意节点和数据据传输错误的判别

每个中间节点在转发数据之前，通过取出 CP_i，计算 $H(CP_i)$。并与包中的值进行比较，若不同，则说明是数据传输错误。由它报告其上一跳，由上一跳生成 ACK 报文，请求源节点重发。若某节点总是收到其下一跳检测出错报告，频度达到某个设定值，则可由该节点报告其下一跳为问题节点。若目的节点收到所有片断，所有片断的 $H(CP_i)$ 值均与计算值匹配，但数据重组时，Reed-Solomon 编码能够发现某个或某些片断包含错误，则可判断传送该片断的路由包含恶意节点，这种判断当传输错误致使计算出的 $H(CP_i)$ 与包中的 $H(CP_i)$ 正好相符时出现误判，其概率为 $\frac{1}{2^{Hash\_Len}}$，$Hash\_Len$ 是散列值的长度，可以认为误判率接近于 0。

4.2 数据篡改恶意节点检测

恶意节点要想修改包中的 CP_i 值，则必须修改其对应散列值 $H(CP_i)$，否则 CP_i 与 Hash 值不匹配，该片断会被中间节点检测出来并丢弃，然后请求源节点重发。当某一路径上多次请求重发，频率超过设定的某个阈值时，会被检测出来。若恶意节点有篡改分组片断，使篡改后的 CP_i 与一个新的 Hash 值相匹配，则中间节点是无法检测到这一恶意行为的。但目的节点在片断重组解码时，是可以发现该错误的。当目

的节点发现有包含恶意节点篡改分组片断后，如图 2 所示，会发起一个检测恶意攻击节点的过程。

图 2　恶意节点 M 篡改片 CP_i 为 CP'_i

设目的节点重组原始分组 CP 后，判断出 CP_i 片断包含错误符号。设收到的与真实 CP_i 对应的篡改后的值为 CP'_i，原节点发出的包中散列值为 $H(CP_i)$，目的节点收到的包中该值为 $H(CP'_i)$，目的节点经纠错后可反向计算出真实的 CP_i，进而可求 $H(CP_i)$，构造检测包，检测包沿出错分组片断所经过的路由的反向传送。设 X 节点的下一跳为 Y，节点 X 的检测包的格式如下：

$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP_i^X))_{K_X^{-1}}, (H(CP_i^Y))_{K_Y^{-1}}]$

其中，

$DETECT$：该报文为攻击检测报文。

SEQ：检测包序号。由节点 D 给出，每发起一次检测，该值加 1，在转发过程中该值不变。

$(CP_i \| CP'_i)_{K_D^{-1}}$：目的节点用私钥加密后的真实和错误的数据片断。

$(H(CP_i^X))_{K_X^{-1}}$：节点 X 用私钥对自己收到的分组片断的 Hash 值加密后的结果。

$(H(CP_i^Y))_{K_Y^{-1}}$：后一节点 Y 用私钥对自己收到的分组片断的 Hash 值加密结果。该值由 Y 发送给 X 节点的检测包中获得。

例如本例图 2 中，由于节点 D 没有后一节点，它所构造传递给前一节点 L 的检测包为：

$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP'_i))_{K_D^{-1}}, (H(CP'_i))_{K_D^{-1}}]$

节点为 L 构造并传递给前一节点 G 的检测包为：$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP'_i))_{K_L^{-1}}, (H(CP'_i))_{K_D^{-1}}]$，节点 G 的检测包为：$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP'_i))_{K_G^{-1}}, (H(CP'_i))_{K_L^{-1}}]$，依此类推。

图 3　检测包的传送过程

若在由 S 到 D 的某条路由中，节点 W 的前一跳为 V，后一节点为 X，X 的后一跳节点为 Y，如图 3 所示，节点 W 收到检测包$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP_i^X))_{K_X^{-1}}, (H(CP_i^Y))_{K_Y^{-1}}]$后，其处理过程如下：

(1) 脱密 $(CP_i \| CP'_i)_{K_D^{-1}}$，$(H(CP_i^X))_{K_X^{-1}}$，$(H(CP_i^Y))_{K_Y^{-1}}$。

(2) 对从 $(CP_i \| CP'_i)_{K_D^{-1}}$ 中脱密出来的 CP_i 和 CP'_i 求取 $H(CP_i)$ 和 $H(CP'_i)$，测试从 $(H(CP_i^X))_{K_X^{-1}}$ 和 $(H(CP_i^Y))_{K_Y^{-1}}$ 中脱密的值是否相同，且与其中之一吻合。如果不是，判定该检测包为非法检测包，将 X 从邻节点表中删除，结束处理。

(3) 从本节点的缓存中查找与 $H(CP_i)$ 或 $H(CP'_i)$ 相匹配的值，记该值为 $H(CP_i^W)$，判断 $H(CP_i^W)$ 是否与 $H(CP_i^X)$ 相等。若相等，转(5)。

(4) 构造举报 ACK 报文，沿该路由分别送源节点 S 和目的节点 D，报告发现错误。该报文经过恶意节点的方向可能会被恶意节点丢弃，但另一方向的传送可以成功到达。因此，举报 ACK 报文总是可以达到节点 S 或 D 的。

(5) 构造检测包$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP_i^W))_{K_W^{-1}}, (H(CP_i^X))_{K_X^{-1}}]$，送前一跳 V，处理结束。

由前提假设，在源节点 S 和目的节点 D 之间，至少存在一条无恶意节点的路由，节点 D 可通过此路由将 $(CP_i \| CP'_i)_{K_D^{-1}}$ 安全送交节点 S，从 S 到 D 的方向，可以同样构成一个检测和举报过程，检测报文格式与上述格式相同，如节点 S 由于它没有前一跳，其构造的检测报文内容为$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP_i))_{K_S^{-1}}, (H(CP_i))_{K_S^{-1}}]$，节点 H 构造的检测报文内容为$[DETECT, SEQ, (CP_i \| CP'_i)_{K_D^{-1}}, (H(CP_i))_{K_H^{-1}}, (H(CP_i))_{K_S^{-1}}]$，其后节点依此类推，检测报文的处理过程与反向检测时检测报文的处理过程完全相同。

经过双向检测，S 和 D 可以综合收到的举报信息，判断出恶意节点位置。由上述处理过程可知，无论恶意节点 M 在构造检测包时用 $(H(CP'_i))_{K_M^{-1}}$ 还是 $(H(CP_i))_{K_M^{-1}}$，在反向检测时，节点 F 向 S 送达举报包，在正向检测过程中，节点 G 会向 D 送达举报，节点 S 和节点 D 综合举报信息后，无论恶意节点是否举报。若两个举报点中间只有一个节点，则可断定该中间节点为恶意节点。这是因为：由前提假设，一条路由中只有不超过一个恶意节点，恶意节点可以在检测过程中生成举报包，也可能不生成，但与之相邻的两个正常节点必然会对其恶意行为进行举报。

恶意节点收到检测报文后，可能并不按上述协议规则处理，其处理方法还存在两种可能：一是丢弃该报文，二是不按协议指定规则构造检测报文转发。下面对这两种情况给出一个预防机制。

4.3　恶意节点不按规则处理检测报文的预防机制

为了防止恶意节点在收到检测报文后，不按协议规则处理，可以采用检测报文传回执的机制，检测报文在传送过程中，被任一节点 X 收到，X 给其发送者一个回执包，由发送者缓存，以备追查。回执包格式为

$[DETECT\_RECEIPT, (H(DETECT\_PKG))_{K_X^{-1}}]$，

其中，

$DETECT\_RECEIPT$：检测包回执报文标识。

$DETECT\_PKG$：收到的检测包。

发送节点在设定的时间间隔内未收到回执报文，则可认为该节点移出或是恶意节点，将该节点从自己的邻节点表中删除。回执报文由于有私钥加密，是不可仿造的。恶意节点若给前一节点回执，但不转发，它将收不到下一节点的回执，这一恶意行为也能被查出。

恶意节点要想篡改检测包，且不被查出，必须使构建的检测包前一跳的 $(H(CP_i^X))_{K_X^{-1}}$ 与 $(CP_i \| CP'_i)_{K_D^{-1}}$ 中的内容，要求在脱密后吻合，恶意节点是不可能做到的。

5　协议能抵制的内部攻击

(1)防止被窃听。由于对分组进行加密后再进行 Reed-Solomon 编码，最后经分片后按不同路径传输，恶意节点窃听密文的 Reed-Solomon 编码片断是无意义的。即使恶意节点获得了源节点和目的节点间的共享密钥，单凭它所截获的部分编码也是无法获得有用消息的。

(2)重放攻击。每个中间节点缓存有前一节点的片断的值的 Hash 值,若出现相同的 Hash 值,则重放报文将会被丢弃。

(3)伪造或者篡改攻击。当一个数据包被篡改,恶意节点修改的只是一个片断,恶意节点本身得不到好处,目的节点可以判断出错误。当错误符号数在 Reed-Solomon 编码的纠错能力范围之内时,目的节点可以重构出正确的原始分组,通过检测过程,最终将其排除在系统之外。当超出纠错能力时,目的节点可请求源节点重发。

6 仿真结果及分析

6.1 仿真环境

实验平台为 Pentium Ⅳ 1.8 GHz,512 MB RAM,使用的操作系统是 Linux 2.4.18,网络仿真平台是 ns2.28(network simulator veReed-Solomonion2.28)。仿真中,节点总数设置为 100 个,节点运动范围 1000m × 1000m。网络中节点的运动方式采用随机运动模型,即每个节点在该区域内从一点向另一点运动,运动速度在零到最大速度之间随机选取。到达目标点后,停留一段时间,然后随机选择一个新的目标点和一个新的速度,向新的目标点运动。依此类推,直至仿真结束。MAC 层使用的 802.11 协议,路由协议选用 AOMDV,在路由寻找期间计算多条路径。节点传输半径为 250m,链路带宽为 2Mbps,在网络中随机开展源节点与目的节点的通信,通信源是连续比特率(Continuous Bit-Rate,CBR)源,数据包长度为 256 字节,模拟时间为 500s。

6.2 部分中断攻击的仿真结果

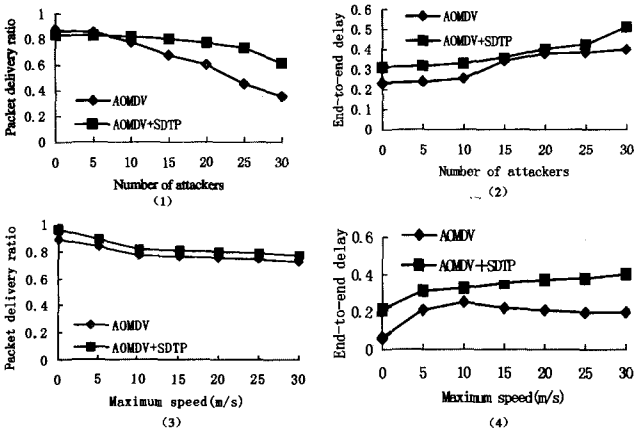

图 4 数据直接传送与通过 SDTP 协议传送的性能比较

图 4(1)-(2)显示了在不同数量的恶意节点攻击下的仿真结果,其中节点的最大移动速度为 10m/s,节点的停留时间为 0s。图 4(1)显示了分组传递率和攻击者数量之间的关系,这里的分组传递率是指分组被正确传送与总分组数的比率。随着攻击者数目的增加,采用 SDTP 传送数据的传递率下降较慢,主要因为攻击者发生恶意行为后,可以被检测出来,从而立即被排除在数据传送路由之外。再者,采用 Reed-Solomon 编码,当数据分片出现错误时,可以被纠正。由于编码和解码具有一定的延时和路由的动态变化,当攻击者数目很少时,采用 SDTP 传送方式的优越性未能表现出来,所以会略低于直接传送方式。当攻击者数目较大时,单条路由中所包含的恶意节点数可能超出一个,不能被及时检测出来。而检测时间会被同样消耗,导致分组传送率下降加剧,但也远高于直接传送时的分组传递率。图 4(2)给出了点到点的延时比较,采用 SDTP 传送方式的延时高于直接传送方式,这是由 Reed-Solomon 编码和解码的延时造成的,是数据在不安全的环境中进行安全传送必然付出的代价。

图 4(3)-(4)显示了在不同的节点移动速度下的仿真结果,其中恶意节点的个数为 10 个,节点的停留时间为 0s。图 4(3)显示出在两种不同数据传送方式下,采用 SDTP 的分组传递率在不同节点移动速度下,均高于直接传送方式。这是因为尽管 SDTP 有编解码的延时,在路由动态变化下会导致分组传递率的下降,但它可以有效抵制恶意节点的攻击,使分组的正确传递率大大提高。图 4(4)显示了两种传送方式下的点到点的延时,采用 SDTP 传送方式的延时高于直接传送方式,SDTP 保证了数据的完全送达,而直接传送方式只完成了分组的传送,但接收方收到的数据是不完整的。

值得说明的是,SDTP 不仅可用于如 AOMDV、多径 DSR 等这些可发现不相交路由的多径路由协议上,而且可用于相交多径路由协议中。在相交多径路由协议中,SDTP 由于可准确定位恶意节点的位置,更可显示其在数据传送时的安全和速度方面的优越性。

结束语 本文基于 Reed-Solomon 编码的纠错技术,提出了一种在存在 Byzantine 攻击的网络环境下的安全数据传输协议 SDTP。该协议不仅能实现数据的安全传输,更为重要的是能定位攻击节点的位置,为 Ad Hoc 网络中检测内部攻击节点提供了一种更准确、具体且实际可行的算法。其优点是在多径路由协议中可以充分利用多条路由提高数据传输的效率和可靠性。SDTP 中采用的恶意节点检测的思想,同样可被用于安全路由协议。由于本文所提之方法局限于讨论在一条路由中只有不超过一个恶意节点情况,当超过一个恶意节点时,在某些情况下该检测方法可能失效,但并不影响其检测结果的正确性和可用性。对于一条路由中可能出现的相邻恶意节点情况下的有效检测方法,是下一步有待更加深入研究的方向。

参 考 文 献

[1] Hu Y-C, Perrig A, Johnson D B. Ariadne: a secure on demand routing protocol for ad hoc networks// Proc. of ACM Mobicom. Atlanta, GA, Sept. 2002

[2] Hu Y, Johnson D, Perrig A. SEAD: Secure efficient distance vector routing for mobile Ad Hoc networks// Proc. of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002). IEEE, Calicoon, NY, June 2002

[3] Dahill B, Levine B N, Royer E, et al. ARAN: A Secure Routing Protocol for Ad Hoc Networks. Umass Tech Report 02-32. 2002

[4] Sanzgiri K, Dahill B, Levine B N, et al. A Secure Routing Protocol for Ad Hoc Networks// 10th IEEE Intl. Conf. on Network Protocols (ICNP02). November 2002

[5] Cheung S, Levitt K. Protecting Routing Infrastructures from Denialof Service using Cooperative Intrusion Detection// Workshop on New Security Paradigms. 1997

[6] Goodrich M T. Efficient and Secure Network Routing Algorithms. http://www.cs.jhu.edu/~goodrich/cgc/pubs/routing.pdf, Provisional patent filing, Jan. 2001

[7] Buchegger S, Boudec J-Y L. Performance analysis of the CONFIDANT protocol// Proc. of the 3rd ACM International Symposium on Mobile ad hoc Networking & Computing. Lausanne. Switzerland, June 2002: 226-236

[8] Buchegger S, Le Boudec J-Y. A robust reputation system for mobile ad-hoc networks. EPFL Technical report. No. IC/2003/50. July 2003

[9] Awerbuch B, Holmer D, Nita-Rotaru C, et al. An Ondemand Secure Routing Protocol Resilient to Byzantine Failures// ACM Workshop on Wireless Security (WiSe). September 2002

[10] Reed I S, Solomon G. Polynomial codes over certain finite fields. SIAM Journal of Applied Math, 1960, 8: 300-304

[11] Reed I S, Chen X. Error-Control Coding for Data Networks. KluwerAcademic PublisheReed-Solomon, 1999

[12] Diffie W, Hellman M E. New directions in cryptography. IEEE Trans. Inform. Theory, November 1976, IT-22: 644-654

[13] 俞波,杨珉,王治,等. 选择传递攻击中的异常丢包检测. 计算机学报, 2006, 29(9): 1542-1552