基于混沌加密的安全 AODV 路由协议研究

杨铭熙,严晓明
(武汉理工大学 计算机科学与技术学院,湖北 武汉 430070)

摘 要:列举了对无线自组网进行攻击的具体方式,比较了国内外研究人员在自组网安全性上的改进方案。考虑到非对称加密方案的计算量与响应时间的开销比较大,而作为流密码的混沌加密又能够满足加密强度和加密时间开销的要求,提出将混沌加密应用到 AODV 路由协议中。描述了对修改后的 AODV 协议的流程,进行安全分析。通过仿真实验分析,该方案在结点数较多的时候,在端到端延时上接近于 AODV。适合于资源有限和对时间开销要求较高的 Ad Hoc 网络。

关键词:混沌;AODV;路由;安全;NS
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2007)05-0107-04

Research of Security AODV Route Protocol Based on Chaos Encryption

YANG Ming-xi, YAN Xiao-ming
(College of Computer Science & Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract: Specialize types of attack on the Ad Hoc, and compare the proposals about the security of the Ad Hoc. Considering the spending in the calculate and respond is great than stream encryption, a scheme of applying Chaos encryption arithmetic into AODV route protocol is proposed. Describe the flow of the AODV after modification and carry out security analysis. Simulation results show that the end to end delay performance is close to AODV when counts of node are sufficient . The scheme fit Ad Hoc that resources limited and high demand of the real time performance.

Key words: Chaos; AODV; route; security; NS

0 引 言

AODV(Ad Hoc on Demand Distance Vector)是一种按需路由协议[1,2],在无线自组网的环境下,仅在有结点需要发送数据,又没有去往目的结点路由的时候才按需发起路由请求,而没有象表驱动一样,不论有无通信需求,都要进行路由信息交换,并且维护去往其他所有结点的路由。网络拓扑结构以及路由信息是按需建立的,只查找和维护自己需要使用的路由,而不是到所有结点的路由。这对于自组网在能量、信道的利用上都具有较好的特性。

作为 Ad Hoc 中的路由协议,安全性也是 AODV 中的一个关键问题。在 AODV 建立路由的时候,需要和其它的结点协作,交换路由信息,没有采取相应的安全措施,很容易成为恶意结点攻击的目标。

1 AODV 的安全性

在对于 Ad Hoc 的攻击上,国内外的许多研究者做了大量的分析[3~6],分成了:拒绝服务攻击(DoS)、路由黑洞(Black hole)、路由重播(Replay)、Rushing 攻击、Tunneling 攻击等这几大类。而这些攻击具体细化到 AODV 协议中,Manel Guerrero Zapata 把恶意结点对 AODV 协议的攻击归纳为以下六类[7]:假冒源结点 S 伪造一个 RREQ 包,让自己的地址看起来象源地址;当转发一个 RREQ 包的时候,减少跳数域;扮演接收方伪造一个 RREP,让自己的地址看起来象目的地址;没有去转发相应的 RREQ 或者 RREP,或者没有去响应相应的 RREQ 和数据包;用很高的目的结点序列号伪造一个 RERR 包;把一个结点的源或目的序列号设置得很大。

针对于无线自组网中的这些攻击,K Sanzgiri 等提出的 ARAN(Authenticated Routing for Ad Hoc Networks)[5],通过可信任的认证服务器,为所有的有效结点颁发证书,并且把路由跳数信息也加入到签过名的包内,下一跳收到路由包的结点,用从密钥服务器中得到的公钥对由上游结点发来的路由包进行验证,通过验证后,再用自己的私钥进行加密后进行转发。在 ARAN 中用到了时间戳,这对于 Ad Hoc 网络的时钟同

收稿日期:2006-07-26
作者简介:杨铭熙(1956-),女,广东人,副教授,研究方向为网络路由技术、网络安全。

步提出了很高的要求；而且对于跳数这个可变的域也与其它不变域进行一样的操作，增加了计算量。Yih-Chun Hu, Adrian Perrig 等提出的 Ariadne[4]，源目结点之间共享密钥，来保证收发双方的身份；并提出了一个哈希算法，逐跳运算保证了中间转发的结点的身份认证。但是，该协议对于路由包中的各结点的地址没有保护的措施，易受到恶意结点的攻击。SAODV[8]提出了把可变域，如跳数，和不可变域分开进行处理，对于可变域，用 Hash 函数逐跳验证，对于不可变域进行签名，并拓展了一些字段用于已知目的结点的中间结点发送 RREP 包。但是由于 SAODV 用到了公钥算法，进行数字签名，使得端到端的开销过大，对于无线自组网这样实时性能要求较高的应用领域而言，有待进一步改进性能。

无线自组网在安全目标上与传统的有线网络的安全目标是一致的，它们包括：认证、存取控制、数据保密性、数据完整性、不可否认性[9,10]。而无线自组网又一般用在军事、救灾等各种需要临时建立通讯网络的场合，在保证安全性能的同时，快速响应也是主要的挑战。以上的这几类相关的安全协议都是建立在公钥的基础上的，而以公钥为基础的算法在时间性能上比对称加密算法来得差。早在 20 多年前，Needham 就指出使用公钥加密算法与使用传统加密算法的协议有着惊人的相似[11]。Olga Kornievskaia 从身份绑定与验证、生存时间与新旧程度、网络实用性、存取粒度上比较了对称与非对称加密算法，细化了 Needham 提出的两类加密算法在以上几个方面上的类似[12]。

2 对称加密的引入

由于非对称加密方案对结点的计算量与响应时间的开销比较大，下文引入对称加密的方式对 AODV 的路由发现机制进行端到端的认证。对称加密算法采用了 Chaos(混沌)加密算法。选择 Chaos 算法主要是基于混沌系统所具有的独特性质：对初值极端的敏感性、密文流具有高度的随机性。

Chaos 可以作为流加密，也可以作为块加密算法来应用。文中应用到混沌的流加密方式，设 K 为初始密钥，M 与 C 分别为明文与密文，Chaos(K)表示把 K 输入到 Chaos 方程中。则在发送方：$C = Chaos(K) \oplus M$；而在接收端：$M = Chaos(K) \oplus C$。

当混沌系统是典型的一维 Logistic 混沌映射：$X_{n+1} = rX_n(1 - X_n)$，$-1 < X_n < 1$ 的时候，选取适当的参数：$r \in (3.57,4]$的时候，Logistic 映射表现出的以上混沌特性可以让输出的密钥流 X_{n+1}在[0,1]区间内不规则分布，将其映射到[0,255]区间内的整数值，再与明文流进行异或得到最终的密文流。当 r 的取值发生变化时，只要是变化任何一个有效位的实数，都将使得最终的密钥流完全不同，混沌的雪崩效应[10]强于其它的对称加密算法(见图 1)。

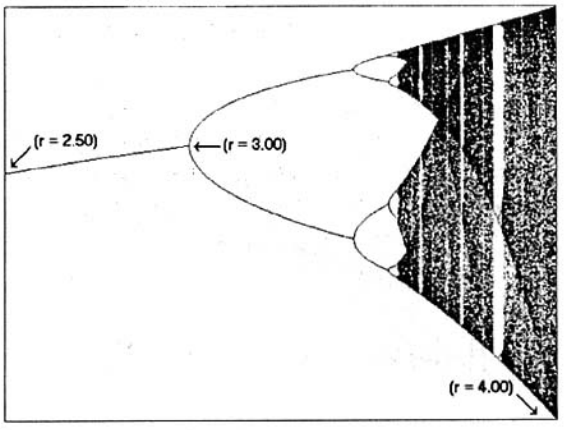

图 1 一维 Logistic 混沌映射

因此，将混沌理论应用于密码学上，具有保密性强、随机性好、密钥量大等特点，并且时间开销小。加密是将明文的信息流变换为可逆的随机流，解密过程则是对数学变的换逆变换处理的过程，将得到的随机流还原为明文。密文的随机性强弱决定了还原为明文的过程的难易程度。而混沌加密的输入密钥的集合为整个实数域，密钥空间大，计算复杂度小。在对于安全性能要求较高的场合，还可以用多个混沌系统进行迭加，加强安全性，但同时会增加时间上的开销。

文中引入的基于 Chaos 的安全路由协议，对于 AODV 路由包中的可变部分(跳数)采用与 SAODV 相同的做法，使用 Hash 函数和 Hash 值进行逐跳认证。在用到哈希链对跳数域进行认证时，对于每一个中间结点都进行一次认证，每次当一个结点要去发送一个 RREQ 或者是 RREP 的时候，它就生成一个随机数；当中间结点收到了这个 RREQ 或者 RREP 的时候，就应用在路由包头部的哈希函数和最大跳数和随机数进行验证；在转发前，再进行重新计算后转发。对于路由包中的不可变部分，采用 Chaos 加密算法进行加密。假设每个结点在加入自组网时，已经从密钥管理结点处得到了相应的密钥，由于 Chaos 本身的特点，这个密钥可以在整个实数域中取值，这个实数由密钥管理结点进行管理，经过认证后的结点，向管理中心发送自己的 ID 及当前路由包的序列号等信息后，可收到中心结点的两个实数，其中一个实数是与上游结点相同的，用于解密；另一个实数可以让这个中间结点更换 Chaos 的密钥，再进行转发，用于和下一跳的结点进行认证。由于密钥管理的安全性，Chaos 方程可以放在 AODV 路

由包的头部进行发送,而且这个路由包也是加过密的,没有相应的密钥管理结点的认证,不能得到这个方程。Chaos 方程的安全性与包中的头部以及包中其它数据一样,全部被经过密钥管理结点确认的实数密钥进行认证,并且每一跳更换一个密钥,极大地增强了路由的安全性能,并且由于 Chaos 为对称加密的特点,在时间性能上也能得到保证。

设源结点 S 要向目的结点 D 发送一个路由请求包 RREQ,包在除跳数(Counts)外的其它数据用 M 来表示,K1 是源结点向密钥管理中心申请的实数,I_i 表示第 i 个中间结点,源结点 S 要申请 K1,I_1 要申请 K1 与 K2,K1 与上游结点(S)的密钥相同,K2 为与下一跳结点共享的密钥,采用逐跳申请密钥的方式,极大地保证了在路由过程中的安全性。发送序列可以表示如下:

$S \rightarrow I_1 : E_{k1}[M \| H(Counts)]$

$I_1 \rightarrow I_2 : E_{k12}[M \| H(Counts')]$

……

$I_{n-1} \rightarrow D : E_{kn-1,n}[M \| H(Counts^{(n-1)})]$

3　安全分析

下面对恶意结点对 AODV 协议的六类攻击分别进行安全分析:

(1)假冒源结点 S 伪造一个 RREQ 包,让自己的地址看起来象源地址。

由密钥管理结点处得到的共有的一对实数作为密钥,只有上下游结点才拥有,这样可以实现路由包的认证,恶意结点 S'没法拥有这个密钥,就不能实现正常的解密,也就不可能去伪造 RREQ。通过这个密钥,也可以实现加密,其它结点没有相应的密钥,也无法解密。对于用 Chaos 进行加密的方式,加密函数放在路由包头部的数据域中,但是密钥(实数)不在域中,恶意结点取得这个包,但是无法从密钥管理结点处得到正确的实数密钥,无法进行解密。

在确定有发送的需要的时候,源分别向密钥管理中心请求实数密钥,各跳分别申请下一跳的密钥,这样,在密钥管理中心有着这一条路径的所有的活动记录,可以实现接收方不能伪造消息,发送方不能否认消息。还可以对恶意结点进行初步的确认。源与目的如果收到消息,就不能进行伪造,因为在密钥管理结点处有双方的请求密钥的记录。如果由于中间结点原因,而造成传输失败的,再由源重新申请实数,重新进行传输过程。

但是当源结点向密钥管理中心再去申请一个密钥时,接收方还不知道有结点要传输消息给自己。那么当到了目的结点的时候,接收方以自己的身份加上传来的 RREQ 包中的上一跳 ID 等数据向管理中心去申请上一跳的密钥,目的结点按与其它结点相同的方式把上一跳的信息给密钥管理中心进行验证,这时,由于密钥管理中心有所有的路径记录,这样就阻止假冒源结点的 S'伪造 RREQ。

(2)当转发一个 RREQ 包的时候,减少跳数域。

采用与 SAODV 相同的 Hash 函数的方法,逐跳验证。

(3)扮演接收方伪造一个 RREP,让自己的地址看起来象目的地址。

当去仿冒 RREP 的时候,由于不知道源结点的实数是什么,这样没有办法去伪造这个实数。

(4)没有去转发相应的 RREQ 或者 RREP,也没有去响应相应的 RREQ 和数据包。

这个是很难被检测到的,因为传输错误也有相同的结果。SAODV 也没有提出相应的解决方法。这里,可以设置源结点向密钥管理中心申请密钥的次数,超过一定的次数,就表明源结点周边存在恶意结点,取消发送。由于对数据进行了加密,其它结点不能去解密。而且这个时候,密钥管理中心就会知道这个结点周边信号差,或者是存在恶意结点,而采取一定的措施。

(5)伪造一个 RERR 包,用很高的目的结点序列号。

可以用与(1)相同的方法来解决。

(6)把一个结点的序列号设置得很大。

不管是源结点还是目的结点的序列号,由于在生成 RREQ 或者响应 RREP 的时候,应用由密钥管理中心得到的密钥进行了加密,可以解决这个问题。

4　仿　真

以上针对 Chaos 的算法拓展的 AODV 协议在 NS2 下进行仿真,并且与采用了 Rc4 加密算法时进行比较,Rc4 是应用最广泛的流密码,被应用于 SSL/TLS 标准,也应用于作为 IEEE802.11 无线局域网标准一部分的 WEP 协议,而且在时间开销上较为合理[10]。

Rc4 加密算法采用 Eric Young 的程序[13],在硬件配置为 Celeron M 1.5G,512DDR,操作系统为 Fedora Core 4 下进行 30 次测试并求出延时的平均值,每加密 1k 的时间为 44μs;Chaos 方程采用一维 Logistic 方程,自编程序进行测试,软硬件环境与 Rc4 加密算法进行测试时相同,也进行 30 次测试并求出延时的平均值,每加密 1k 的时间为 22μs。

仿真过程中,设置了一个 1000m×1000m 的区域,仿真时间设为 400s。分别在 20 到 60 个结点下进行测

试,采用 CBR 数据源,设定发送速率为每秒 1 个数据包,结点最大移动速度为 40m/s, CBR 数据源根据仿真场景中的结点数的不同,保持在 60% 左右的连接率,数据分组的长度为 512 字节。对于不同的结点数,分别进行了 10 次的仿真,每次仿真设置了不同的 seed。并算出场景中端到端延迟时间的平均值。对于不同结点数应用了 Chaos 与 Rc4 对称加密的端到端延迟进行了统计(如图 2 所示)。

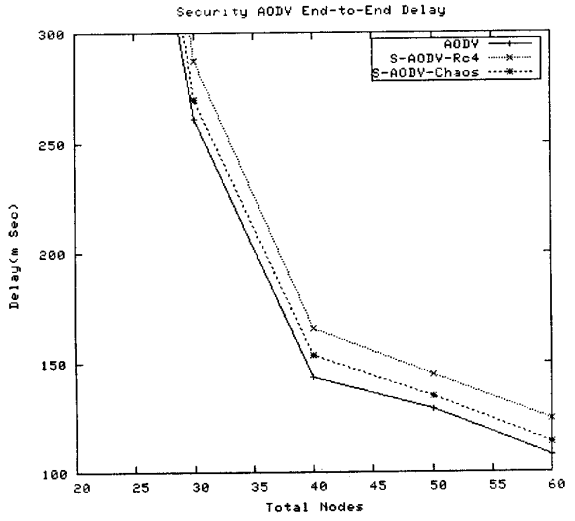

图 2 不同结点数下的端到端延迟

从图中可以看到,随着结点数的增加,原 AODV 协议与使用了对称加密算法进行加密的安全 AODV 协议的端到端延迟差值逐渐减少。其中,使用 Chaos 进行加密比用 Rc4 进行加密具有更好的时间性能。

5 总结与下一步的工作

在对 AODV 的加密方案中引入混沌加密算法,在提高 AODV 安全性能的同时,对于时间上的开销也能控制在一个较小的范围内,适合对于时间开销要求较高的场合。目前各个对于 AODV 的安全性能的研究中,对于密钥管理结点的可靠性以及性能要求较高,而相关的一些研究都没有展开,因此对于密钥管理结点的研究是下一步要进行的重要的工作。

参考文献:

[1] Ad hoc On - Demand Distance Vector (AODV) Routing[EB/OL]. 2003. http://www.ietf.org/internet - drafts/draft - ietf - manet - aodv - 13.txt.

[2] Perkins C, Royer E. Ad hoc on demand distance vector routing [C] // In: The 2nd IEEE Workshop on Mobile Computing System and Applications. New Orleans, LA: [s.n.], 1999.

[3] Papadimitratos P, Haas Z J. Secure Routing for Mobile Ad hoc Networks[C]//CNDS 2002. San Antonio, TX: [s.n.], 2002: 27 - 31.

[4] Hu Yih - Chun, Perrig A, Johnson D B. Ariadne: A Secure On - Demand Routing Protocol for Ad Hoc Networks[R]. Department of Computer Science, Rice University, 2001.

[5] Sanzgiri K, Dahill B. A secure routing protocol for Ad Hoc networks[C] // Proceedings ICNP2002. Paris, France: IEEE Computer Society, 2002: 78 - 89.

[6] 王英龙. 移动 Ad Hoc 网络的安全路由协议研究[C]// DPCS2003 & NDCS13 论文集. 大连:大连理工大学,2003.

[7] Zapata M G, Asokan N. Securing Ad hoc Routing Protocols [C]//WiSe'02, 2002. Atlanta, Georgia, USA: [s.n.], 2002.

[8] Secure Ad hoc On - Demand Distance Vector(SAODV)Routing[EB/OL]. 2005. http://www.potaroo.net/ietf/all - ids/draft - guerrero - manet - saodv - 05.txt.

[9] Zhou L, Haas Z J. Securing Ad Hoc Networks[J]. IEEE network, 1999, 13(6): 24 - 30.

[10] Stallings W. Cryptography and Network Security: Principles and Practices[M]. 3rd edition. [s.l.]: Prentice Hall, 2003.

[11] Needham R, Shroeder M. Using encryption for authentication in large networks of computers[J]. Communications of the ACM, 1978, 21(12): 993 - 999.

[12] Kornievskaia O. Symmetric and Asymmetric Authentication: A Study of Symmetric and Complementary Properties and Their Effect on Interoperability and Scalability in Distributed Systems[EB/OL]. 2002. http://www.citi.umich.edu/u/aglo/papers/thesis-proposal.pdf.

[13] Young E. libRC4. tar.gz[CP/OL]. 1997. ftp://ftp.psy.uq.oz.au/pub/Crypto/.

(上接第 106 页)

患于未然,采取有效措施,避免感染后再消除,甚至是造成无法挽回的损失。防重于杀,是今后防病毒的必然要求。

参考文献:

[1] 周公望. 浅析计算机病毒[J]. 计算机与网络,2005(2):42 - 45.

[2] 马亚丽. 浅析计算机病毒的预防和清除[J]. 甘肃农业,2005(2):82 - 83.

[3] 周 琴. 计算机病毒研究与防治[J]. 计算机与数字工程,2006,34(3):86 - 90.

[4] 李向宇,郭 薇. 计算机反病毒技术实用指南[M]. 北京:国防工业出版社,1992.

[5] 黄毅静. 计算机病毒知识及其反病毒技术[J]. 山东省青年管理干部学院学报,2003(2):108 - 109.

[6] 韩同跃,曲忠庆,刘东彦. 两种反病毒模式的对比研究[J]. 洛阳师专学报,1998,17(5):36 - 38.