IPv6 邻居发现过程的安全性研究

明廷堂
河南省开封市河南大学网络中心(475004)
E-mail:mingtingtang@126.com

摘要：
邻居发现是一种专用于 IPv6 的新协议。最初的邻居发现过程的定义是基于本地链路由相互信任的节点组成的理想情况。但是，无线网络（如无线局域网）技术的发展，深刻地改变了这一假设。在本地链路上的节点不能再理所当然地相互信任，甚至 在节点通过网络完成了身份验证，节点依然对对方产生怀疑。这产生了一系列的可操作性难度和安全威胁。

本文对针对 IPv6 中的邻居发现过程的安全威胁进行分类，并描述两种新的加密方法，即加密产生地址和基于密钥的地址。同时探讨如何使用这些方法来加强 IPv6 中的邻居发现过程的安全架构。

关键词：邻居发现、加密地产生地址、基于密钥的地址

1 概述

在最初提出 IPv6[1][2]基础设计时，很难预测像今天一样需要考虑使用 IPv6 的各种无线环境。相应地，维护本地链接的 IPv6 功能被设计成物理保护，信任连接是自然的。现在，人们开始考虑在公共场合应用无线网络，比如在机场、旅馆等场合架设无线局域网（WLANs）。但是，对于无线网络，因为它采用无线连接，通过电磁波的收发实现数据的传输，在任何电磁波有效覆盖区域范围内的用户都有可能与现有网络进行连接。尽管无线连接在第二层也存在一定的身份验证、访问控制、信息加密（如 IEEE 802.1x 和 802.11i），链路上的一些节点依然是不可信任的。在无线网络应用中，WLAN 是整个无线网络的焦点，尽管它在安全性与其它无线网络相比有非常强的保证，各项安全标准也比较完善，但与传统的有线网络相比，其在安全性方面仍然显得苍白无力，特别是对于那些专门进行网络数据窃取的黑客。

本文不对整个 IPv6 的安全性进行论述，专注于 IPv6 中的邻居发现过程[4]的安全性研究。它们的当前定义依赖于本地链路上不存在不被信任的节点。事实上，一个非常简单的不被信任节点都可以发起很多攻击。尽管当前针对 ND 的 RFCs[3][4][5][6]进行了一定的补充，但是没有提供关于如何使用安全架构 IPSec 来加强 IPv6 安全性的更多细节，更重要的是在 ND 中使用 IPSec 会产生一系列的问题。

2 背景知识

本小节我们将简明地介绍当前的 ND 技术中的一些要点。

2.1 IPv6 邻居发现过程

邻居发现（Neighbor Discovery） 是一种新的协议，简称 ND 协议。节点（主机和路由器）使用它来确定相同网络和附加链路上邻居的链路层地址。ND 协议

组合了 IPv4 协议 ARP、ICMP 路由器发现和 ICMP 重定向等。ND 协议用来查找转发数据报的邻居路由器，跟踪邻居的可达性状态，并检测新的和更改的链路层地址。当邻居失效时，还可以使用邻居发现表来快速的确定是否存在可替换的路径。

ND 协议定义在 RFC 2461，它被用来解决关于相同物理链路上的节点之间的交互问题。RFC 2461 定义了所有的 ND 交互操作，包括：路由器发现、前缀发现、地址自动配置、重复地址检测、邻居不可达性检测、链路层地址解析、下一跳确定以及重定向。

邻居发现过程使用五个不同的 ICMPv6 数据报类型来确定和维护 IPv6 路由器之间的邻居关系，它们是路由器请求、路由器通告、邻居请求、邻居发现、重定向。所有这些消息都承载在 ICMPv6 消息中。

路由器请求数据报（Router Solicitation，RS）的 ICMPv6 消息类型为 133。在启用 IPv6 接口来请求来自邻接路由器的中间路由器通告时，RS 将被发送到所有路由器组播地址。RS 使得邻接 IPv6 路由器可以用一个路由器通告消息来响应，这就使得主机能够立即自动配置其接口。

路由器通告数据报（Router Advertisement，RA）的 ICMPv6 消息类型为 134。RA 被定期发送到所有节点组播地址来通告它们的存在，或者被发送来响应路由器请求消息。这种通告通常包含本地链路节点用来自动配置其 IPv6 地址的前缀、每个通告前缀的生存期消息、表示无状态或有状态自动配置的标志、发送该通告的路由器是否应该被用作默认路由器，以及一些其它主机信息（比如跳数限制和 MTU 等）。

邻居请求数据报（Neighbor Solicitation，NS）的 ICMPv6 消息类型为 135。NS 将被发送到请求节点的组播地址，并确定相同链路上邻居的链路层地址。邻居请求还可以被发送到邻居的单播地址，用来验证邻居的可达性。或者用于重复地址检测 NS 消息使得邻接 IPv6 节点可以用一个邻居通告消息来响应。

邻居通告数据报（Neighbor Advertisement，NA）的 ICMPv6 消息类型为 136。NA 被发送来响应邻居请求消息。邻居通告消息和发送它的 IPv6 接口源地址一起被发送。邻居请求的发送者收到邻居通告之后，两个节点皆可以通信。节点还可以发送未请求的邻居通告来宣告链路层地址的更改。

2.1.1 IPv6 地址解析（Address Resolution）

为了学习其它节点的链路层地址，需要启动地址解析过程。源节点向目标节点的组播地址发送一个邻居请求消息 NS。如果目标节点存在，它将会侦听该组播地址。一旦目标获取邻居请求报文，就发送一个邻居通告消息 NA 来响应源节点。相关信息携带在 NS 和 NA 报文中。此交互过程中使用 IPSec AH 来保护 ND 消息。

但是，从安全性考虑，IPSec AH 因为密钥分发问题并未起到事实上的保护作用，在身份验证的背后隐藏更多的问题。比如，NA 消息包含许多的标志域，除非身份验证信息跟 IP 地址捆绑，否则目标节点没有办法确信通过数据包验证的源节点是否真的被授权声明为该源地址的宿主。

2.1.2 IPv6 邻居不可达性检测（Neighbor Unreachability Detection，NUD）

链路上的节点通过邻居不可达过程检测链路上的目标节点[4]。通常节点依据更高层的信息来确定对端节点是否可达。如果在更高层的数据流有足够长的延

迟或目标节点中止接收数据，NUD 过程就被激活。源节点首先等待小段随机延迟。然后向目标节点发送 NS 报文，如果对端依然可达，它就发送 NA 响应源节点；如果请求节点没有收到响应，它就等待一段稍长的延迟后，从邻居缓存表中删除该目标节点。必要时，这会触发地址解析过程。

2.1.3 IPv6 重复地址检测（Duplicate Address Detection，DAD）

当一个节点获取一个新的地址时，它首先必须确信链路上没有其它的节点使用该 IP 地址。这就触发 DAD 过程。源节点通过向本地链路发送一系列的 NS 消息来完成。这些消息包含源节点需要使用的试探性 IP 地址。如果源节点没有收到回应，它就可以使用该地址；如果该地址已被占用，正在使用的节点将会发送一个 NA 消息回应源节点，源节点就开始尝试新的 IP 地址。

2.1.4 路由器发现（Router Discovery，RD）

邻居发现过程允许主机同本地链路上的节点通信，同时节点也需要学习链路上的路由器的标识与能力。为此，RD 过程为节点提供全局的可路由的地址前缀。通常，本地链路上的所有路由器周期性地组播 RA 消息，为了标识发送路由器，每个 RA 报文包含一系列的可路由前缀——节点就可以据此获取全局的可路由地址和一条默认路由。当然，节点也可以初始化一个 RD 过程，通过发布 RS 消息，期待本地路由器进行通告回应。

2.1.5 重定向（Redirect）

在重定向过程中，路由器使用重定向数据报来通知主机到达目的地的最佳首跳点。重定向用于路由优化。没有重定向，路由器依然会建立与维护通信，并通过其它的路径转发数据包。

重定向消息通常从一个单播地址发送到触发重定向过程的数据包的源地址[4]。当目的地址不是组播地址时，或当数据报没有被指定发送到该路由器，或当数据报将被接受它的接口发送出去时，或当数据报的源地址是相同地址或者本地链路地址（Link-local Address）上邻居的全局 IPv6 地址时，路由器就使用重定向数据报。在 ND 交互的所有消息中，仅重定向消息用于链接-本地而不是端到端通信。

2.1.6 自动配置（Auto Configuration）

无状态自动配置规范[5]定义了一种 IP 主机在启动时即可初始化配置的方法。在基本的无状态自动配置过程中，正在启动的主机首先执行 DAD 过程，试图获取一个链接-本地地址；一旦主机获得一个链接-本地地址，就进入自动配置的第二个阶段，启动 RD 过程，试图从路由器那里获得地址和前缀以及一条默认路由；最后，主机需要知道本地的 DNS 服务器。

3 ND 安全威胁模型

在开放式网络环境中，由于任何节点都可以不需通过链路层验证就可以加入到本地连接，其中就可能隐藏着恶意的节点。除非在节点与网络设备之间存在某种安全性上下文，否则节点是不可信任的。因此，ND 的安全性研究成为一个日益重要的课题。

下面讨论的许多的威胁与攻击在 IPv4 网络中普遍存在。随着 IPv6 的日益广泛的应用，在 IPv6 中提及这些问题也是必要的，特别是在公共的无线网络环境。

一般地，有两种类型的威胁：

- 各种拒绝服务攻击（DoS）。目的一般是为了使目标主机无法再正常提供服务。被成功攻击的主机因网络过载而速度缓慢，直至瘫痪。
- 重定向威胁。攻击者将数据包从最后一跳的路由器重定向到链路上另外的节点。重定向攻击可以用于 Dos 攻击，也可以用于其它的攻击。

重定向攻击用于三种主要目的：

1、数据包被探测和截取。有时是在几乎不可能的场合，如交换局域网中。

2、数据包被重定向到一个错误或不存在的链路层地址，阻止地址的真正宿主不能正常通信。

3、大量的数据被重定向到一个存在的链路层地址，阻塞网络接口，引起主机的处理器和日志文件崩溃。

通常，地址宿主验证是一种行之有效的应对重定向攻击的手段。

3.1 重定向威胁

3.1.1 恶意的最后一跳路由器

在本地链路上的恶意路由器能重定向所有流过它的数据包[4]。攻击者通过广播合法的路由器通告消息或单播响应多播路由器请求消息伪装成最后一跳路由器。如果主机选择攻击者作为默认路由器，攻击者就有机会从该主机截取信息流。同时，攻击者一旦成为合法的路由器，它就可以向主机发送重定向消息。

3.1.2 NS/NA 欺骗

通过从一个伪造的源链路层地址发送 NS 消息或者向一个伪造的目标链路层地址发送 NA 消息，攻击者可以为合法的节点（包括路由器和主机）伪造数据包，并发送到某些合法的链路层地址。如果该伪造地址有效，只要攻击者发送响应 NUD 的 NS 消息的时间足够长，数据包就会持续的被重定向。

这种模式可通过指定一个未使用的链路层地址用于 DoS 攻击，但是，由于 NUD 过程在一定的时间间隔（默认值 30 秒）内会丢弃非法的地址，攻击的时间收到限制。因此，如果攻击者想持续攻击，它就必须使用捏造的地址持续响应。

3.1.3 伪造重定向消息

重定向消息能用于将一个给定目标 IP 地址的数据包重定向到链接上的任何链路层地址。攻击者可以使用首跳路由器的链路-本地地址作为源地址向合法的主机发送重定向消息，由于主机认识来自它的首跳路由器的链路层地址的消息，主机就会接受重定向。只要攻击者响应 NUD 来探测链路层地址的时间足够长，重定向就会持续生效。

3.2 DoS 威胁

3.2.1 伪装的 On-Link 前缀

攻击节点能发送一个 RA 消息，指明任意长度的前缀处于 On-Link 状态（源与目的在同一链路上）。如果主机认为该前缀是 On-Link 的，它就永远不会向路由器发送该前缀数据包。事实上，主机会尝试发送 NS 消息执行地址解析过程，但是不会接收到 NA 响应，导致主机拒绝服务。

攻击者可以利用该伪装的前缀通告一个时间段。如果该时间段无限制，主机就会一直拒绝服务直到它从本机的前缀列表中删除该前缀。

3.2.2 伪装的自动地址配置前缀

攻击节点能发送一个错误的 RA 消息并指明主机使用一个自动配置的无效的子网前缀。当主机执行地址自动配置算法时，它就使用该通告前缀来构建自己的地址[5]，即使该地址在该子网中是无效的。因为源地址无效，返回数据包永远不会到达主机。

3.2.3 DAD 拒绝服务攻击

在主机通过无状态地址配置[5]获得接口地址的网络环境中，一个攻击节点能在 DAD 过程中发起 DoS 攻击。如果攻击者声明拥有地址，主机就永远不能获得地址。这种威胁在 RFC 2462[5]中指出。

3.2.4 邻居发现拒绝服务攻击

在这种模式中，由于最后一跳路由器通过 ND 协议进行地址解析，攻击节点使用目标网络的子网前缀伪造地址并持续向它发送数据包。合法的主机尝试进入网络可能永远不能从最后一跳路由器获取 ND 服务，因为路由器忙于解析伪造的地址。不同的是这种 DoS 攻击的节点可能是 Off-Link（源与目的不在同一链路上）的。该模式中被攻击的主要资源是邻居缓存，它被填充满了需要解析的地址——这些地址的前缀有效而尾部却无效。

3.3 IPSec 密钥管理中存在的问题

为了降低上述攻击的风险，加强安全性，邻居发现交互过程中的消息使用 IPSec 验证头（AH）[4]保护机制。潜在地，定义合适的 AH 安全关联（SA）集，主机可以精确的验证 NA 和 RA 消息。一种关于通过手工配置定义必要的 SA 集的方法已经提出。然而，基于下列原因，当前除了手动配置外还没有其它的机制提供这样一个必要的 SA 集：

- 当前唯一可行的自动创建 SA 的方法是 IKE。为了执行功能，IKE 需要一个 IP 堆栈。为了维护 IP 堆栈，在 IKE 中就存在引导问题。
- 即使通过一些其它的可行的非手动方式来建立必要的 SA 集，也是对验证动态生成或分配地址的归属问题于事无补。

4 加密地产生地址

4.1 CGA 基本思想

CGA 基于这样一种思考：使用 IPv6 地址的低 62 比特来存储一个公开密钥的哈希加密序列。基本算法可以表达如下：

$$\text{Host ID}=\text{HASH62}(\text{public\_key}) \qquad (4\text{-}1)$$

其基本思想是将一个地址同一个公开密钥充分地绑定在一起。在某些场合，不使用公开密钥加密而又能声明地址的归属具有很大的优势。为此，CGA 生成一个哈希链：

$$H_N = \text{HASH}_{160}(\text{public\_key}|\text{random})$$
$$H_i = \text{HASH}_{160}(\text{public\_key}|\ H_{i+1})$$
$$\text{Host ID} = \text{HASH}_{62}\ (H_0) \qquad (4\text{-}2)$$

主机就不需要公开密钥加密就可以产生哈希散列$H_0,\ldots, H_N$。在冲突方面，由于哈希值H_i可以但是不必局限于 62 比特（比如 160 比特），冲突的可能性微乎其微。

在基本的 CGA 中，对于暴力攻击来说，特别是攻击者使用大量的地址寻找

冲突的攻击中，要获得足够强的安全性和真正的保护，62 比特还显得不够。但哈希链方法没有改变哈希长度的有效性，还需要其它的手段。

4.2 突破 62 比特限制

一个有效提高哈希长度的方法是将安全参数嵌入到地址中。该参数决定散列函数使用的哈希长度。该安全参数——*Sec*,是取 128 比特 IPv6 地址的低 2 比特的无符号整数编码。尽管这使得散列函数仅有 60 比特作为输出，但是获得了更优的性能。

在修正的模式中，从公开密钥中计算出一个 128 比特的哈希值算法为：

$$HashValue=HASH(public\_key) \quad (4\text{-}3)$$

一般地，产生一个 CGA 地址可以按照下列步骤：

1、选择安全参数 *Sec*=0，1，2，3；
2、生成一个合适的哈希输入集；
3、根据哈希输入执行哈希算法；
4、提取哈希输出的低 64+20×*Sec* 位，并将该低 64+20×*Sec* 位中的高 20×*Sec* 位同 0 比较，如果值不为 0，跳到第 2 步重新执行；
5、连结 64 比特路由前缀和哈希输出的低 64 位形成一个 128 位的 IPv6 地址；
6、设置该地址的组和全局标志位均为 1，低两位设置成安全参数；
7、如果检测到地址冲突，跳到第 2 步重新执行；如果三次连续冲突，就停止执行算法并报告错误。

使用这种模式，创建一个新的CGA的代价取决于安全参数*Sec*，*Sec*可以取值 0，1，2，3。对于一个给定的安全参数，算法在主机找到输入集对应的哈希输出中低 64+20×*Sec*位中的高 20×*Sec*位为 0 的输出时终止。*Sec*=0 时，能根据哈希输入及直接算法计算出一个合适的哈希值，并嵌入到地址中，在第 4 步中的零比较中，成功率很高；*Sec*=1，2，3 时，对于一些好的哈希加密算法，输出是随机分布的，为了在第 4 步的零比较中成功获取合适的哈希输入，主机必须尝试 $2^{(20\times Sec-1)}$个不同的输入。

4.3 CGA 地址绑定

上面的过程中，仅仅输入公开密钥通过散列的方法产生主机标识符。事实上，还可以将其它的数据迭加在哈希输入上。

通过将CGA地址与某个网络地址或者硬件地址进行绑定，可以缓和暴力攻击。为此，仅需将网络路由前缀或者主机的链路层地址作为哈希输入。如果它们都迭加在哈希输入上，攻击者将要以极大的代价对 2^{60}或 2^{62}个可能的接口标识符中的每一个计算一个包含合适的密钥对的回环链表。该回环链表用来声明任何IP地址的归属。将路由前缀或硬件地址包含在CGA中，由于攻击者需要为每个网络地址或接口创建一个独立的表，这需要攻击者付出巨大的性能代价。考虑当前攻击者处理器和存储系统的因素，针对CGA保护的攻击似乎不可能，从而保护系统。

但是，一方面，数据处理和存储能力不断飞速得到提高，在 IPv6 的生命周期中，针对 CGA 保护的攻击将会变得可行。另一方面，对于每个路由前缀或网络接口，CGA 必须使用不同的标识符，这也是它的一个不足。

可以运用相同的技术，将与安全相关的数据同 CGA 地址绑定。

5 基于密钥的地址

基于密钥的地址（ABK）使用知名的基于加密机制的标识技术。基于加密机制的身份认证允许任何公认的标识符（比如 E-mail 地址、节点 IP 地址）或任何比特字符串充当加密密钥/解密密钥对中的加密密钥。技巧在于产生对应的解密密钥和安全参数。ABK 使用 IP 地址作为加密密钥。

基于加密机制的标识技术为加密社区熟知已近 20 年。但是，它没有被网络安全社区广泛交流、探讨。一个主要的原因是基于 Diffie-Hellman 算法的加密标准。另外，它一直缺乏一个高效的产生基于加密的标识的算法。目前，这种形势已经改变。

基于加密机制的标识技术执行流程：一个公开的标识符被提交到一个可信任的机构 IPKG。IPKG 使用独特的算法产生解密密钥并通过安全隧道返回。这种模式在异步加密系统中广泛应用于认证和加密。在密钥产生过程中，IPKG 使用特定的参数集——包含一个只有 IPKG 知道的主控参数和其它的为依赖 IPKG 的合作者所知的参数。

5.1 基于密钥的标识算法

存在许多可行的基于加密机制的标识算法。比如基于信号量的标识模式[21][22]、基于密钥协商的标识模式、基于加密的标识模式。

在基于信号量的标识协议中，主机使用 IPKG 支持的秘密密钥发布信号量，信号量被主机所公开的标识符进行验证；在基于密钥协商的标识协议中，二者共享秘密信息，双方都使用自己的秘密密钥和对方的公开密钥构建秘密信息；在基于加密的标识协议中，加密者使用接收者公开的标识符加密信息，接收者使用自己的秘密密钥对加密信息解密。

5.2 计算数字信号量

ABK 中的数字信号里量使用下列算法：

sig=SIGN(HASH(contents),IPrK,Parameters) (5-1)

其中：sig 是数字信号量；SIGN 是基于数字信号的标识算法；HASH 是一种哈希算法，比如 SHA1-HAMC；IPrk 是基于秘密密钥的标识；Parameters 是公开的加密参数；contents 是发布的消息内容。

接收者通过下列方法验证该信号量：

IPuK=IBC-HASH(ID) (5-2)

valid=VERIFY(HASH(contents),sig,IPuK) (5-3)

其中：IBC-HASH 是专用于基于从公开的标识符产生公开密钥的标识算法的哈希函数；ID 是用于产生公开密钥的公开的标识符；IPuK 是基于公开密钥的标识；sig 是数字信号量；VERIFY 是用于验证信号量的基于公开密钥的标识算法；valid，布尔值，1 表示通过验证，0 表示未通过验证。

6 CGA 和 ABK 在加强 IPv6 邻居发现过程的安全性中的应用

CGA 能用来验证来自地址宿主（如产生地址的节点）的回答。这可以通过为包含产生地址的密钥的消息打标记做到。这种技术能用于 ND 交互中的地址解析、重复地址检测、重定向。而 ABK 同样具备这些功能。一方面，CGA 不需要可信任的第三方机构；另一方面，CGA 需要强制选择 IP 地址而 ABK 不需要。

因此 CGA 在接口标识符受到限制的场合不宜使用。

6.1 CGA 加强 IPv6 地址解析安全性

一个加强地址解析安全性的方法是将 CGA 地址对应的公开密钥和信号量包含在地址解析交互过程使用到的所有 NS 和 NA 消息中。如果其它的参数也用于产生 CGA 地址，它们也必须包含其中发送。接收者能根据这些信息进行验证。收到 NA 消息的节点能立即验证地址，并将消息存储在缓存中以备以后使用。

验证者首先重新计算公开密钥的哈希值并将结果同接口标识符进行比较；然后，验证者验证使用公开密钥的消息中的信号量。如果二者验证都通过，验证者就可以将该地址添加到地址解析缓存，并丢弃验证数据。

6.2 CGA 加强 IPv6 重复地址解析安全性

因为仅仅 CGA 地址的宿主能用可验证的信号量进行响应，所有针对 DAD 查询的欺骗尝试都将受到攻击保护。

在 IPv6 无状态地址自动配置中，可能会检测到地址冲突并尝试新的地址。然而地址冲突几乎不可能发生，除非有针对该协议的攻击发生。连续三次冲突的情况更是罕见，如果偶然发生，可以肯定的是存在某种攻击。

6.3 CGA 加强 IPv6 重定向安全性

路由器可以使用 CGA 来证明重定向消息来自声明的 IP 地址。主机使用它进行验证，确认该 IP 地址同已用于下一跳的地址相同。这其中没有涉及重定向过程中的其它路由器。但是，可以交叉验证路由器，而让主机验证重定向源地址和重定向目的地址是否在交叉验证节点集中。这使得潜在的重定向攻击变得十分困难。

6.4 ABK 加强 IPv6 路由发现过程安全性

如上所述，CGA 在 ND 的一些交互过程中是一种很好的安全机制，但是，对于 ND 中的路由器发现过程，还不够。相反，ABK 可以传达关于第三方授权信息。

为了加强 RA 安全性，在路由器上，ABK 可用来标记和授权可路由前缀。事实上，64 比特的子网前缀可以作为公开密钥而秘密密钥可以从相同的比特衍生。该秘密密钥应用于计算一个包含在 RA 消息中的信号量。因此，原则上，基于信号量的 ABK 参数必须包含在所有的 RA 消息中。为了安全，该模式中的公开的 ABK 参数必须安全地指明以便网络中的其它节点能验证信号量。建议 ABK 参数与秘密密钥使用相同的安全协议。路由器的 ABK 参数和秘密密钥可以手工配置。

为了加强 RS 的安全性，主机有两种途径使用 ABK：

1、当主机执行安全的第 2 层身份认证与授权过程获取 IP 地址并进入网络时，由网络为主机提供 ABK 参数和秘密密钥。

2、主机随着秘密密钥预配置其 ABK 参数。

为了授权可路由前缀，RA 消息需要携带一个简单的前缀选项，表明该前缀的密钥已经分配。如果路由器需要路由其它前缀，它需要用独立的 RA 进行通告。

一个 IPv6 主机接收到一个承载有 ICMPv6 信号量选项的 RA 消息时，它使用下面方法验证该通告节点是否被授权发送该通告：首先，主机确定简单的可路由前缀选项并提取 RA 发送节点声明的允许路由的子网前缀；然后，主机使用

ABK 算法验证信号量。此计算过程中，ABK 公开密钥是选项中的子网前缀。

因此，如果主机信任公开密钥的 ABK 参数，它就能验证发送 RA 消息的节点是否真的拥有与该路由前缀对应的秘密密钥。因为此类秘密密钥只有在合法的路由器上支持有效，信号量就能很好的证明 RA 消息是否完整并且是否由合法的路由器发送。

7 总结

本文介绍了当前与 IPv6 邻居发现过程相关的一些安全威胁，并简要的描述了两种新的加密技术 CGA 和 ABK 以及如何应用 CGA 和 ABK 加强 IPv6 发现过程的安全性，同时也指出 CGA 和 ABK 的一些局限性。

参考文献

[1]S.Deering,R.Hinden. Internet Protocol Version 6(IPv6) Specification,RFC2460. IETF,December 1998.

[2]A.Conta,R.Hinden. Internet Control Message Protocol(ICMPv6) for the Internet Protocol Version 6(IPv6) Specification,RFC2463. IETF,December 1998.

[3]S.Deering,R.Hinden. IP version 6 Address Architecture,RFC2373. IETF Network Working Group,July 1998.

[4]T.Narten,R.Draves,W.Simpson. Neighbor Discovery for IP version 6(IPv6),RFC2641. IETF,December 1998.

[5]S.Thomson,T.Narten. IPv6 Stateless Address Autoconfiguration,RFC2462. IETF,December 1998.

[6]T.Narten,R.Draves. Privacy for Stateless Address Autoconfiguration in IPv6,RFC3041. IETF,January 2001.

The Study of Neighbor Discovery Security in IPv6

Tingtang MING

Network Center , Henan University,Kaifeng,Henan,475004

E-mail:mingtingtang@126.com

Abstract:

Neighbor-Discovery is a new protocol for IPv6.When IPv6 Neighbor Discovery functions were defined, it were assumed that the local link would consist of mutually trusting nodes. However, the recent development in public wireless networks, such as WLAN, have rapidly changed the situation. The nodes on a local link cannot necessarily trust each other any more, but they must become mutually suspicious even when the nodes have completed an authentication exchange with the network. This creates a number of operational difficulties and new security threats.

In this paper we provide a taxonomy for the IPv6 Neighbor Discovery security threats, describe two new cryptographic methods, Cryptographically Generated Addresses(CGA) and Address Based Keys(ABK),and discuss how these methods can be used to secure the Neighbor-Discovery mechanisms.

Key：Neighbor Discovery(ND)、Cryptographically Generated Addresses(CGA)、Address Based Keys(ABK)