



基于 MT6D 的物联网轻量级安全协议研究

黄廷辉, 丁 勇, 李思骏

(桂林电子科技大学 计算机与信息安全学院, 广西 桂林 541004)

摘 要: 随着物联网设备在开放互联网上的快速部署, 物联网设备的隐私和通信安全问题得到广泛关注。由于物联网的嵌入式设备受到资源和计算性能的限制, 传统的网络通信安全方法难以提供可靠的安全保障。为此, 提出一种基于低功耗无线个域网 6LoWPAN 的轻量级 IPv6 地址跳变协议 (L6HOP)。通过对移动目标 IPv6 网络防御 (MT6D) 协议进行改进, 使用轻量级哈希算法降低 CPU 计算消耗, 并引入滑动地址窗口解决不同设备时钟误差引起的丢包率较大的问题。实验结果表明, L6HOP 协议可有效保护物联网不受设备追踪、DoS 和窃听等攻击, 与 MT6D 协议相比, 可有效减少 CPU 的计算开销, 并能够降低通信丢包率。

关键词: 物联网; 地址跳变; 安全协议; 移动目标 IPv6 网络防御协议; 通信效率

开放科学 (资源服务) 标志码 (OSID):



中文引用格式: 黄廷辉, 丁勇, 李思骏. 基于 MT6D 的物联网轻量级安全协议研究[J]. 计算机工程, 2020, 46(9): 117-122.

英文引用格式: HUANG Tinghui, DING Yong, LI Sijun. Research on MT6D-based lightweight security protocol for Internet of things[J]. Computer Engineering, 2020, 46(9): 117-122.

Research on MT6D-based Lightweight Security Protocol for Internet of Things

HUANG Tinghui, DING Yong, LI Sijun

(College of Computer and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China)

[Abstract] With the rapid deployment of Internet of Things (IoT) devices on the open Internet, the privacy and communication security of IoT devices has attracted much attention. Because the embedded IoT devices are limited by resources and computing performance, traditional network communication security methods have been unable to provide reliable security guarantee. Therefore, this paper proposes a lightweight IPv6 address hopping protocol, L6HOP, in the low-power wireless personal area network, 6LoWPAN. The protocol improves the Moving Target IPv6 Defense (MT6D) protocol, and uses a lightweight hash algorithm to reduce CPU computing consumption. Also, the sliding address window is introduced to solve the high packet loss rate caused by clock errors of different devices. Experimental results show that the L6HOP protocol can effectively protect the IoT from device tracking, DoS and eavesdropping attacks. Compared with MT6D protocol, it can significantly reduce the computing overhead of CPU and packet loss rate of communication.

[Key words] Internet of Things (IoT); address hopping; security protocol; Moving Target IPv6 Defense (MT6D) protocol; communication efficiency

DOI: 10.19678/j.issn.1000-3428.0055664

0 概述

物联网通常由低功耗、资源受限的无线设备组、传感器和控制器等通过互联网进行连接和通信, 被广泛应用在通信、医疗、交通等多个不同领域。预计到 2020 年底, 物联网连接设备的数量将达到 300 亿^[1]。随着物联网的发展和提供服务的增多, 其受到网络攻击的风险也随之增加。物联网设备的隐私和通信安

全问题受到人们的广泛重视, 由于现有的安全协议存在安全漏洞和高能耗等缺陷, 因此研究一种能保证物联网安全且不影响其通信效率的低功耗协议是目前急需解决的问题。

近年来, 研究人员针对物联网通信安全进行了大量研究。文献[2-3]分别提出入侵检测系统以监控恶意活动, 但由于入侵检测系统计算需求大, 无法应用到低功耗、低资源的设备中。目前, 有学者已经

基金项目: 国家自然科学基金 (61662016); 赛尔网络下一代互联网技术创新项目 (NGII20160306)。

作者简介: 黄廷辉 (1970—), 男, 副教授、硕士, 主研方向为物联网安全、分布式计算; 丁 勇、李思骏, 硕士研究生。

收稿日期: 2019-08-05 修回日期: 2019-09-12 E-mail: 1703303028@mails.guet.edu.cn

研究出许多其他方案,如文献[4-5]提出的身份认证方案、文献[6-7]提出的通信数据加密方案等,但是实施加密和身份验证方案并不会掩盖设备的地址,从而给攻击者提供了查找和定位这些设备并进行攻击的机会。

采用地址跳变的移动目标防御技术(Moving Target Defense,MTD)可以限制攻击者查找和跟踪目标,保护设备的位置隐私,直接影响攻击者的攻击能力。文献[8]开发了一种移动目标 IPv6 网络防御(Moving Target IPv6 Defense,MT6D)协议,该协议通过不断改变网络层和传输层地址,成功地阻止了目标网络、主机跟踪和窃听等攻击,保护了通信双方的隐私和安全^[9]。文献[10-11]提出将 MT6D 用于保护物联网网络的安全,并验证了其方案的可行性,但未考虑物联网设备资源的有限性和通信设备之间的时钟差异导致丢包率较大的问题。文献[12]给出了 uMT6D 协议的概念,即对 MT6D 协议在物联网上的应用进行优化,并提出可以通过 Contiki 系统的 cooja 平台对其方案进行仿真测试,但文献[12]仅提出相关概念而没有具体实现方法。

针对 MT6D 协议在物联网的实现计算开销较大和丢包率较高的问题,本文提出 L6HOP 协议,该协议采用轻量级哈希函数替代 MT6D 中计算量大的 SHA256 算法来降低 CPU 计算开销,并引入滑动地址窗口减小通信设备之间的时钟误差,以保证通信效率,实现物联网设备的主动网络防御。

1 MT6D 协议

通常由于主机不经常更改网络地址或保持静态网络地址不变,攻击者可以窃听通信双方的交互信息或攻击特定的主机。文献[8]针对上述问题并根据军事上无线通信中跳频思想提出 MT6D 协议,以维护 IPv6 网络用户隐私并防范目标网络攻击。IPv6 的引入能解决目前 IPv4 地址数量不足的问题,但由于 IPv6 的无状态地址自动配置(Stateless Address Auto Configuration,SLAAC)^[13]允许设备创建 IPv6 地址,且其创建地址中的主机接口标识符 IID^[14](Interface ID)由 MAC 地址生成,恶意用户可以通过 IP 地址识别设备监视和跟踪这些设备的通信^[15-16]。跟踪和监视一个地址可以使攻击者在侦察阶段有足够的时间收集信息,并最终规划出一个攻击策略。MT6D 协议利用 IPv6 子网中巨大的地址空间动态改变设备 IPv6 地址,通信双方均根据该协议不断改变本地主机地址,并计算对方主机的 IP 地址以保持通信。该协议可以保护主机在公共互联网上相互通信时不受定向网络攻击以及保持通信主机的匿名性。其地址更改是通过改变主机接口标识符 IID 实现的,IID 计算过程如式(1)所示:

$$\text{IID}'_{X(i)} = H[\text{IID}_X \parallel K_S \parallel t_i]_{0 \rightarrow 63} \quad (1)$$

其中,IID'_{X(i)}表示主机 X 在 t_i 时刻跳变后的 IID,IID_X

表示主机 X 当前时刻的 IID, K_S 表示共享的对称密钥, t_i 表示第 i 次计算跳变 IID 的时间, $H[\cdot]$ 表示进行哈希计算,在该协议中哈希算法采用的是 SHA256, $H[\cdot]_{0 \rightarrow 63}$ 表示取哈希值的最左 64 位。IID_X初始值为主机 X 的初始接口标识符 IID,改变后的 IID 由哈希值最左边的 64 位(0 位~63 位)构造。MT6D 通过将主机的子网前缀与 IID'_{X(i)}连接起来形成新的 IPv6 地址。该协议除了动态改变 IID 外,还改变网络端口号,并提供两种动态改变端口号的技术:一种是主机指定一个接近于正常网络流量的端口范围;另一种是使用式(1)中哈希值未使用的位来改变端口号,即若主机可用端口范围为 0~65 535,则使用式(1)中哈希值的第 64 位至第 79 位(共 16 位)来改变端口号。

MT6D 协议使用式(1)在每次到达时间增量时重新计算每个通信对的发送方和接收方的地址,文献[8]测试该协议的可行性,为了便于流量的分析,设置时间增量为 10 s,即该协议每隔 10 s 重新计算通信双方的 IPv6 地址。

2 L6HOP 协议设计

本文针对 MT6D 协议在物联网上应用时未考虑到通信双方设备存在时钟误差和低功耗设备资源受限的问题,提出一种 L6HOP 协议。该协议选择轻量级的哈希函数 Spongint^[17]替代 MT6D 协议中的 SHA256,减少物联网设备计算开销,并引入滑动地址窗口机制解决通信设备之间的时钟误差,降低通信丢包率。

2.1 地址生成

基于 MT6D 协议地址计算过程,本文为增加攻击者破解地址跳变规律的难度,保留了计算当前地址的完整哈希值 h_i ,同时为了避免通信双方主机时钟不同步,不对时间戳参数 t_i 进行哈希计算,则有:

$$h_i = H[h_{i-1} \parallel K_S] \quad (2)$$

$$\text{IID}_{X(i)} = (h_i)_{0 \rightarrow 63}, i = \{1, 2, \dots\} \quad (3)$$

其中, h_i 表示 t_i 时刻的哈希值, h_i 初始值 h_0 为设备 X 的初始 IID,IID_{X(i)}表示主机 X 在 t_i 时刻跳变后的 IID, $(h_i)_{0 \rightarrow 63}$ 表示对数据 h_i 进行取前 64 位操作, $H[\cdot]$ 表示进行哈希计算。针对 MT6D 协议计算消耗较大的问题,文献[18]提出可使用轻量级的哈希函数替代计算量大的 SHA256。文献[19]针对轻量级哈希进行研究,在 ATMEL AVR ATtiny45 8 位微控制器上实现 SHA256、Spongint 等不同哈希算法,并对不同哈希函数的性能从代码大小和内存消耗方面进行比较,实验结果表明,Spongint 函数代码量大小约为 SHA256 的 1/2,内存消耗约为 SHA256 的 1/3,因此式(2)中的哈希计算采用 Spongint 函数替代 MT6D 协议采用的 SHA256 函数。在协议初始化过程中,通信双方采用线下或内部网络传输的方式交换主机的初始 IID 和密钥 K_S ,避免攻击者窃听和篡改数据,保证初始化过程的安全性。由此,地址变化后的 IPv6 地

址表示为 $\text{Addr}_i = \{\text{子网前缀}, \text{IID}_{X(i)}\}$, 定义 IPv6 地址链表为 $\{\text{Addr}_0, \text{Addr}_1, \text{Addr}_2, \dots\}$ 。通信双方根据该算法计算下一跳地址, 网络端口号可根据哈希值 h_i 中未使用的位进行动态改变。

2.2 地址链表更新

本文协议地址链表状态的更新由基于时间的计数器 N_{now} 的更改触发, 计数器 N_{now} 表示地址更改的次数。 N_{now} 的计算公式如式(4)所示:

$$N_{\text{now}} = \left\lfloor \frac{T_{\text{now}} - T_0}{\Delta t} \right\rfloor \quad (4)$$

其中, T_{now} 为实时时钟提供的当前时间, T_0 为通信设备初始连接的时间戳, Δt 表示时间步长, 即地址跳变的时间间隔, $\lfloor \cdot \rfloor$ 表示对数据进行向下取整, N_{now} 表示从开始时间 T_0 到系统当前时间 N_{now} 之间地址变更次数。定义 N_{stored} 保存 N_{now} 的当前值, N_{now} 和 N_{stored} 初始值均为 0。在通信过程中, 当计算结果 $N_{\text{now}} > N_{\text{stored}}$ 时, 则令 $N_{\text{stored}} = N_{\text{now}}$, 并同步更新地址链表; 当计算结果 $N_{\text{now}} = N_{\text{stored}}$ 时, 则不改变当前地址链表信息。

本文所有时间变量均以秒为单位进行度量。由于地址更新是根据时间间隔步长计算的, 因此在设定好时间步长 Δt 之后, 即便客户机上的 T_0 值与服务器上的 T_0 值不同, 也不会影响地址跳变的同步。

2.3 滑动地址窗口

通信双方在通信过程中由于网络时延和时钟差异等, 可能会出现以下 2 种情况: 1) 设备节点的时钟

较快, 而服务器时钟较慢, 此时设备节点已经跳转到下一地址, 导致服务器无法连接到设备跳变前的地址; 2) 设备节点的时钟较慢, 而服务器时钟较快, 设备节点地址还未跳转, 导致服务器无法连接设备节点的下一地址。

为提高通信效率, 降低通信双方因时钟误差引起丢包率较大的问题, L6HOP 协议引入滑动地址窗口机制, 如图 1(a) 所示, 发送地址窗口大小为 1, 定义一个大小为 w 的接收地址窗口并根据 L6HOP 会话的安全需求设置窗口 w 的大小, 地址窗口随着时间增长向前滑动。接收地址窗口即 2.1 节中所定义的地址链表, 地址链表长度由接收地址窗口大小决定。本文设置接收地址窗口大小为 3, 则通信主机每次会话保存发送方地址链表中的 3 个地址。定义当前连接的地址为 Addr_i , 保存的地址即为 $\{\text{Addr}_{i-1}, \text{Addr}_i, \text{Addr}_{i+1}\}$ 。如图 1(b) 所示, 在 t_4 时刻, 当发送方由于时钟太快而先跳转到下一地址, 接收方会在接收窗口选择下一个地址与发送方保持通信; 如图 1(c) 所示, 在 t_4 时刻, 当接收方时钟太快而达到地址跳变条件, 不会立即改变通信地址, 而是保持当前连接的地址不变, 接收窗口向前滑动一次, 等待发送方的当前地址生命周期结束再跳转到下一个地址。滑动地址窗口机制的引入, 保证了每个 L6HOP 协议会话在前后一个时间步长大小范围内的小时钟漂移都能保持正常连接, 确保了连接的灵活性和连续 IPv6 地址之间的平稳过渡。

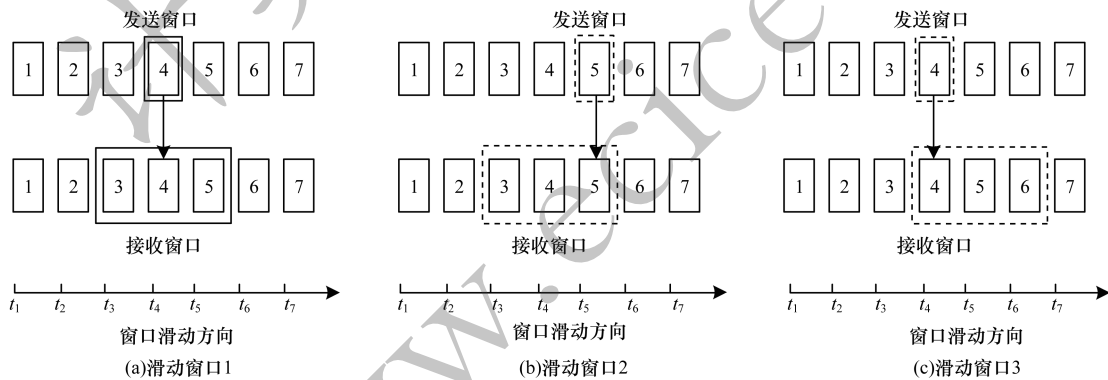


图 1 滑动地址窗口

Fig.1 Sliding address window

2.4 实现过程

L6HOP 协议分别应用在服务器和每个物联网节点上, 物联网系统中服务器与物联网节点的通信为一对多通信, 所以本文只改变物联网节点的 IP 地址, 服务器地址保持不变。在初始化过程中将物联网节点的初始 IID 和密钥 K_s 通过线下传输或者内部网络发送给服务器; 物联网节点的发送窗口大小为 1, 无须设置接收窗口; 服务器分别为每个物联网节点设置一个大小为 w 的接收窗口, 无须设置发送窗口。当物联网节点计数器 N_{now} 发生更改时达到地址跳变条件, 物联网节点

利用该协议计算下一跳地址并更新地址链表, 由于发送窗口大小为 1, 因此链表更新即实现设备 IP 地址的跳变。当服务器计数器 N_{now} 更改时, 服务器利用该协议计算与其通信的物联网节点的下一跳地址, 更新地址链表, 并利用滑动地址窗口机制保持与节点的通信效率。

3 安全性分析

文献[8-11]均设定地址跳变时间为 10 s 时对协议进行分析, 本文参考文献[8-11], 设定 L6HOP 协议以每隔 10 s 动态改变物联网设备的 IPv6 地址来

分析其安全性。当节点设备地址随时间动态改变时,攻击者无法从 IPv6 地址中获得设备的 MAC 地址或者其他静态标识来追踪设备的通信,保证了物联网设备通信的匿名性,能有效防止主机跟踪和窃听攻击,从而实现对设备安全和数据隐私的保护。下文分别对 L6HOP 协议的抗流量截获分析能力和抗 DOS 攻击能力进行分析。

3.1 抗流量截获攻击能力

IPv6 地址后缀为 64 位,它的子网地址为 2^{64} 个,地址空间巨大,远程攻击者想要准确地获取某个目标节点的 IPv6 地址进行攻击的可能性极低。即使攻击者在其最有利的地方,即路由器处对设备的通信子网进行流量分析,也无法确定目标节点的 IP 地址。设子网内通信节点的个数为 L ,通信时长为 S 秒,则在 S 秒内地址跳变总数为 N_{Total} 次, N_{Total} 计算公式如式(5)所示。如图 2(a)所示,子网内实际通信节点个数为 $L = 3$ 个,地址跳变的时间间隔 $\Delta t = 10$ s,则通信时长为 $S = 100$ s 时攻击者探测的地址情况如图 2(b)所示,得到的地址数为 $N_{\text{Total}} = 30$ 个,攻击者截获的是分散的不同节点的通信流量,无法确定截获的是否为目标节点的数据包。

$$N_{\text{Total}} = \frac{S}{\Delta t} \times L \quad (5)$$

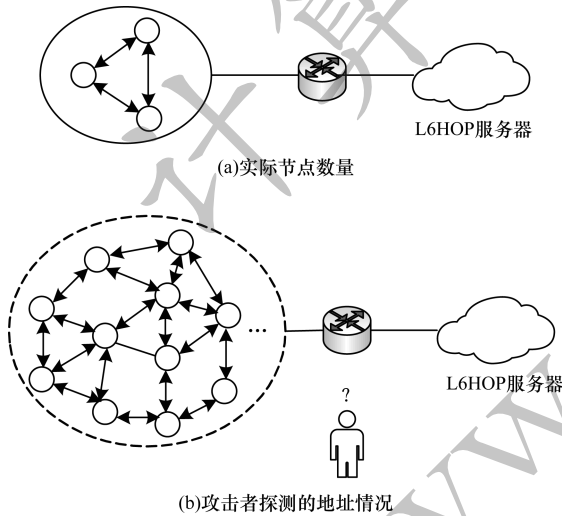


图 2 L6HOP 协议概念示意图

Fig. 2 Conceptual diagram of L6HOP protocol

若攻击者希望通过截获数据包计算出地址跳变的规律,必须计算出共享密钥和 2 个相邻地址生成的完整哈希值。攻击者通过截获的数据包分析得到 2 个连续的地址,分析出其中某个地址 64 位地址后缀的完整哈希数和密钥,需对相邻的两个地址进行计算并比较,至少需要进行 H_{Total} 次哈希计算,则有:

$$H_{\text{Total}} = 2 \times m \times 2^{n-64} \quad (6)$$

其中, n 表示哈希计算生成的哈希值的位数, m 表示攻击者计算共享密钥需要的计算次数。L6HOP 协议的

哈希值 h_i 的位数为 128 位,即使攻击者已知共享密钥也至少需要进行 $H_{\text{Total}} = 2 \times 2^{128-64}$ 次哈希计算。以目前计算机的计算能力无法计算出该协议的地址跳变规律。综上所述,该协议能抵抗流量截获攻击。

3.2 抗 DoS 攻击能力

在实际生活环境中,攻击者可以通过 DoS 攻击达到消耗设备资源或者使设备陷入瘫痪的目的。设备加入 L6HOP 协议,从而不断改变 IP 地址,攻击者无法判断地址跳变规律,不能通过正常的静态 IP 对目标节点进行 DoS 攻击。攻击者可以采用对目标网络的设备节点发送大量地址的数据包进行 DoS 攻击,但是 IPv6 地址跳变的空间为 2^{64} 个,要保持对节点 DoS 攻击百分之百的成功率,每一时刻至少需要发送 2^{64} 个报文到目标网络的子网内。这需要庞大的分布式拒绝服务 (Distributed Denial of Service, DDoS) 攻击才能实现,攻击代价远大于收益,因此, L6HOP 能有效抵抗 DoS 攻击。

4 实验结果与分析

本文在 Contiki3.0 系统的 cooja 仿真平台和在 32 KB RAM、512 KB flash 的 CC2538SF53 芯片的开发板上均验证了跳变协议 MT6D 和 L6HOP 的可行性和有效性,以 Intel i7-7700 CPU、8 GB 内存的个人电脑为服务器与动态地址跳变的 6LoWPAN 网络通信。由于 RPL^[20] (Routing Protocol for Low-Power and Lossy Network) 边界路由器是 6LoWPAN^[21] 网络节点与互联网通信的唯一网关,因此本文将一个节点设定为 RPL 边界路由器,其他节点设定为普通地址跳变的节点。当普通节点地址跳变后只有将新的 IPv6 地址告知 RPL 边界路由器并将该地址加入路由表,地址跳变后的节点才能与互联网正常通信。为避免新地址加入路由表产生延迟,设定普通节点在每次地址跳变后发送一个 RPL 信息对象消息 DIO (DODAG Information Object),并禁用 Contiki 系统中设定的请求消息 DIS (DODAG Information Solicitation) 和确认消息 DDA (DODAG Destination Advertisement) 之间的延时函数。本文设定的网络前缀为 bbbb:0000:0000:0000。

4.1 L6HOP 协议的有效性验证

通过以上环境设置分别测试服务器与 cooja 仿真的单个节点进行 Δt 为 5 s、10 s 和 15 s 的通信,并在服务器上利用 Wireshark 流量分析软件对节点的网络进行流量分析,记录其地址跳变情况,结果如图 3 所示。从图 3 可以看出,节点的地址随着通信时间的增加在不断改变,不同 Δt 的地址跳变频率不同, Δt 越小地址变化越快,但同一节点的地址变化规律相同。实验结果表明,目标节点通信被不同的地址有效分散,证明了 L6HOP 协议能有效保护设备的隐私和通信安全。

$\Delta t=5\text{ s}$	$\Delta t=10\text{ s}$	$\Delta t=15\text{ s}$	地址变化规律
0 s~5 s	0 s~10 s	0 s~15 s	bbbb:: 65E9 : 482F : 3979 : F050
5 s~10 s	10 s~20 s	15 s~30 s	bbbb:: EE8F : 00A4 : 3BA2 : F8B7
10 s~15 s	20 s~30 s	30 s~45 s	bbbb:: A6EC : F31F : 6E27 : 83C7
15 s~20 s	30 s~40 s	45 s~60 s	bbbb:: FA5F : 43AD : F7E1 : 8330
20 s~25 s	40 s~50 s	60 s~75 s	bbbb:: EABE : AF36 : 9FB7 : C697
25 s~30 s	50 s~60 s	75 s~90 s	bbbb:: 84C4 : 8966 : C0EA : 7846
30 s~35 s	60 s~70 s	90 s~105 s	bbbb:: BB4B : 1617 : E552 : B538
35 s~40 s	70 s~80 s	105 s~120 s	bbbb:: F52A : 9FA3 : BD5C : BF4A
...

图 3 地址跳变结果

Fig. 3 Result of address hopping

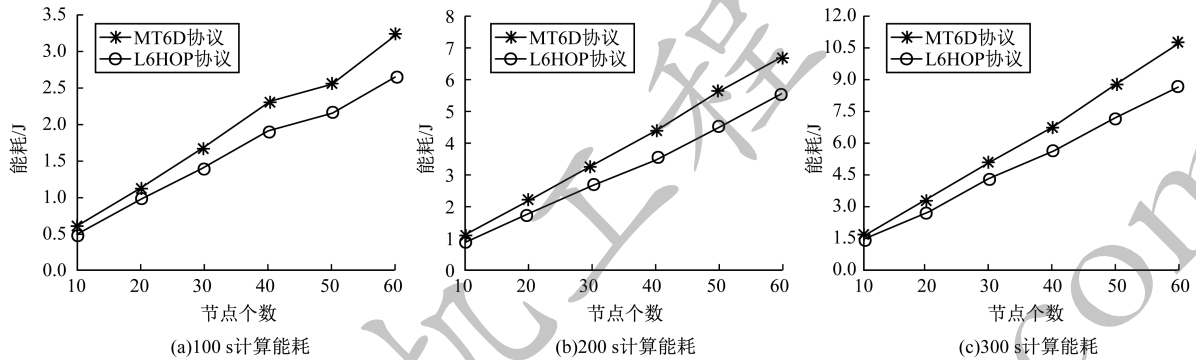


图 4 CPU 计算能耗比较

Fig. 4 Comparison of CPU computing energy consumption

4.3 丢包率对比

为了验证通信双方在时间差异影响下丢包率大小的情况,本文在 cc2538 芯片的节点上分别实现 L6HOP 和 MT6D 两个地址跳变协议,通过服务器向每个节点每秒发送 50 个大小为 10 B 的数据包,并通过 Wireshark 软件抓包分析 2 个不同协议的丢包率, Δt 设为 10 s。由于在实际环境中存在同一个域网下有多个通信节点,因此分别测试 1 个节点、2 个节点和 3 个节点同时与服务器通信的丢包率情况,结果如图 5 所示。从图 5 可以看出,不同数量的节点与服务

器通信对丢包率基本没有影响;MT6D 协议随着通信时间的增长丢包率呈逐渐上升状态,300 s 内上升约 4%;L6HOP 协议能保持较低的丢包率,随着通信时间的增长丢包率基本不变。由于设备节点时钟与服务器的时钟不一致导致服务器与设备节点在时间计算上存在差异,且时间越长差异越大,因为 MT6D 协议下一跳地址是根据时间增量触发,时间差异会导致发送方和接收方地址跳变不同步,时间越长丢包率越大。而 L6HOP 协议滑动地址窗口机制的引入能容忍时间差异,降低丢包率,保持通信效率。

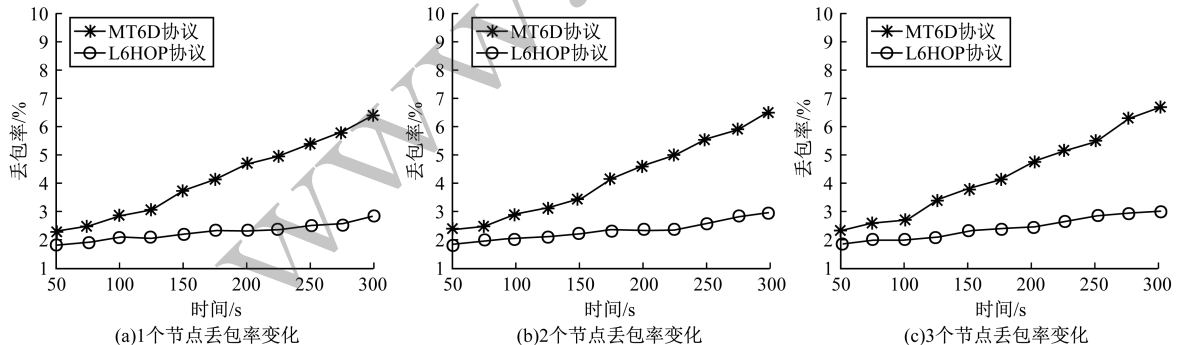


图 5 不同数量节点丢包率比较

Fig. 5 Comparison of packet loss rate of different number of nodes

5 结束语

本文针对 MT6D 协议用于保护物联网安全时计算消耗较大和丢包率较高的问题,提出一种基于低

功耗无线个域网 6LoWPAN 的轻量级 IPv6 地址跳变协议(L6HOP)。通过使用轻量级哈希函数 Spongnet 降低计算消耗,并引入滑动地址窗口机制解决通信双方的时间差异问题。该协议限制攻击者成功识

别 IP 地址从而在目标节点上启动恶意攻击的机会,能够有效抵抗流量截获分析、主机跟踪、窃听攻击和 DOS 攻击等,保护了物联网设备隐私和通信安全。在 cc2358 芯片上的实验结果表明,L6HOP 协议在 CPU 计算开销和丢包率上相比于 MT6D 协议均具有优势。下一步将把该协议与轻量级身份认证技术进行融合,以更全面地保护物联网安全。

参考文献

- [1] PENG Anni,ZHOU Wei,JIA Yan, et al. Survey of the Internet of things operating system security[J]. Journal on Communications,2018,39(3):22-34. (in Chinese)
彭安妮,周威,贾岩,等. 物联网操作系统安全研究综述[J]. 通信学报,2018,39(3):22-34.
- [2] MUDGERIKAR A,SHARMA P,BERTINO E. E-spion: a system-level intrusion detection system for IoT devices[C]//Proceedings of the 2019 ACM Conference on Computer and Communications Security. New York, USA: ACM Press, 2019:493-500.
- [3] ARSHAD J,AZAD M A, ABDELLATIF M, et al. COLIDE: a collaborative intrusion detection framework for IoT[J]. IET Networks,2018,8(1):3-14.
- [4] AKBARZADEH A, BAYAT M, ZAHEDNEJAD B, et al. A lightweight hierarchical authentication scheme for Internet of things[J]. Journal of Ambient Intelligence and Humanized Computing,2019,10(7):2607-2619.
- [5] SHARMA G, KALRA S. A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications[J]. Journal of Information Security and Applications,2018,42:95-106.
- [6] MENG Qian, MA Jianfeng, CHEN Kefei, et al. Data comparable encryption scheme based on cloud computing in Internet of things[J]. Journal on Communications,2018,39(4):167-175. (in Chinese)
孟倩,马建峰,陈克非,等. 基于云计算平台的物联网加密数据比较方案[J]. 通信学报,2018,39(4):167-175.
- [7] RAJESH S,PAUL V,MENON V G, et al. A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices[J]. Symmetry,2019,11(2):293.
- [8] DUNLOP M,GROAT S, URBANSKI W, et al. Mt6d: a moving target IPv6 defense[C]//Proceedings of Military Communications Conference. Washington D. C., USA: IEEE Press,2011:1321-1326.
- [9] DUNLOP M, GROAT S, URBANSKI W, et al. The blind man's bluff approach to security using IPv6[J]. IEEE Security and Privacy,2012,10(4):35-43.
- [10] PREISS T, SHERBURNE M, MARCHANY R, et al. Implementing dynamic address changes in contikios[C]//Proceedings of IEEE International Conference on Information Society. Washington D. C., USA: IEEE Press, 2014: 222-227.
- [11] SHERBURNE M, MARCHANY R, TRONT J. Implementing moving target IPv6 defense to secure 6LowPan in the IoT and smart grid[C]//Proceedings of the 9th Annual Cyber and Information Security Research Conference. New York, USA: ACM Press,2014:37-40.
- [12] ZZITE K, CANTRELL M, MARCHANY R, et al. Designing a micro-moving target IPv6 defense for the IoT[C]//Proceedings of the 2nd IEEE/ACM International Conference on Internet-of-Things Design and Implementation. Washington D. C., USA: IEEE Press,2017:179-184.
- [13] SHAH J L, PARVEZ J. Optimizing security and address configuration in IPv6 SLAAC[J]. Procedia Computer Science,2015,54:177-185.
- [14] CARPENTER B, JIANG S. Significance of IPv6 interface identifiers[J]. Internet Engineering Task Force,2014,65:1-10.
- [15] DUNLOP M, GROAT S, MARCHANY R, et al. The good, the bad, the IPv6[C]//Proceedings of the 9th Annual Communication Networks and Services Research Conference. Washington D. C., USA: IEEE Press,2011: 77-84.
- [16] GRANJAL J, MONTEIRO E, SILVA J S. Security for the Internet of things: a survey of existing protocols and open research issues[J]. IEEE Communications Surveys and Tutorials,2015,17(3):1294-1312.
- [17] NABEEL N, HABAEBI M, MUSTAPHA N C, et al. IoT light weight crypto functions[J]. International Journal of Interactive Mobile Technologies,2019,13(4):117.
- [18] ZEITZ K, CANTRELL M, MARCHANY R, et al. Changing the game: a micro moving target IPv6 defense for the IoT[J]. IEEE Wireless Communications Letters, 2018,7(4):578-581.
- [19] BALASCH J, EGE B, EISENBARTH T, et al. Compact implementation and performance evaluation of hash functions in attiny devices[C]//Proceedings of International Conference on Smart Card Research and Advanced Applications. Berlin, Germany: Springer, 2012:158-172.
- [20] YÜ Ke, WANG Huifeng. Research and Improvement of RPL Routing Protocol[J]. Computer Engineering,2018, 44(3):103-108. (in Chinese)
俞柯,王慧峰. RPL 路由协议的研究与改进[J]. 计算机工程,2018,44(3):103-108.
- [21] PAI V, SHENOY U K K. 6LowPan—performance analysis on low power networks[C]//Proceedings of International Conference on Computer Networks and Communication Technologies. Berlin, Germany: Springer, 2019:145-156.

编辑 索书志