Resilient against spoofing in 6LoWPAN networks by temporary-private IPv6 addresses

Article *in* Peer-to-Peer Networking and Applications · January 2020

DOI: 10.1007/s12083-019-00792-6

2 authors:

Monali Mavani
Birla Institute of Technology and Science Pilani
15 PUBLICATIONS **95** CITATIONS

SEE PROFILE

Krishna Asawa
Jaypee Institute of Information Technology
52 PUBLICATIONS **289** CITATIONS

SEE PROFILE

Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-019-00792-6

Resilient against spoofing in 6LoWPAN networks by temporary-private IPv6 addresses

Monali Mavani[1] · **Krishna Asawa**[1]

Received: 12 November 2018 / Accepted: 11 July 2019

Abstract

An attacker can disrupt the network operations in the 6LoWPANs by spoofing the IPv6 address while evading the detection. Despite many existing spoofing prevention techniques, spoofing threat still persists. Thus, it becomes necessary to devise a method which can offer resilience against spoofing by reducing the attack disruption time. This study aims at reducing IPv6 spoofing attack disruption time in 6LoWPANs. Hence, it provides the resiliency against IPv6 spoofing threat. The time complexity analysis of the attack tree for the spoofing attack is performed to analyze the attack disruption time. The analytical results show that attack disruption window is directly proportional to the lifetime of the node addresses. The lower lifetime of node addresses ensure the reduction of the attack disruption window. Thus, the use of temporary node addresses can be a solution for reducing the spoofing attack disruption window. Node's IPv6 address can be changed periodically to dissociate a node from its permanent identity. Hence, an attacker has to re-perform the attack to gain significant benefits. Corrupted routing table as a result of spoofing attack and its countermeasure is simulated in Cooja running Contiki operating system. The length of the attack window depends upon the periodicity of the address change. The higher frequency of address change decreases the attack disruption time with an increase in the communication cost. Simulations have been performed to compare the optimum value of address change periodicity concerning the communication cost for two private addressing schemes proposed in the literature.

Keywords IPv6 spoofing · 6LoWPAN · Time-To-Live · Attack disruption window · Privacy addressing

1 Introduction

IPv6 over Low Power Personal Area Network (6LoWPAN) has emerged as a leading networking protocol to connect resource constrained nodes to the Internet. Diversified applications which use 6LoWPAN protocol may be deployed in unattended or physically insecure environments. There is a possibility of malicious insider nodes due to the absence of robust node authentication mechanisms [33]. In such environments, spoofing in the wireless radio medium becomes possible. Spoofing itself can be an attack, or it is special in that it can be used to carry out further attacks like DoS, DDoS, Man In Middle attack, Impersonation attack, disrupting routing by corrupting routing tables etc. [4]. When spoofing itself is an attack, it can prohibit a victim to do specific actions. For example, messages from an attacker that spoof the identity of a legitimate node may cause a neighbour relationship to form and prohibits the victim from forming the relationship with the legitimate neighbours. In both the cases, an attacker can disrupt the network as long as its presence goes undetected. This study is motivated to devise the technique which can reduce the attack disruption time due to spoofing in the 6LoWPAN networks. The attack disruption time can be reduced by periodically changing the node's addresses. This study revisits the attack tree defined in [24] to achieve the said objective and incorporates parameters like *Attack Disruption Window (ADW)* and *Time-To-Live(TTL)* to the attack tree. Attack tree modelling gives the opportunity to analyse attacks concerning different parameters such as attack disruption time, time to perform the attack, attack feasibility etc. Attack tree of [24] has modelled the spoofing attack on IP-MAC binding, but it lacks analysis which helps in finding out temporal characteristics of the attack. Temporal analysis by finding time complexity of the attack enables deriving *TTL* and *ADW*. An attacker can perform

✉ Monali Mavani
monamavani@gmail.com

[1] Jaypee Institute of Information Technology, Noida, India

Published online: 14 August 2019

disruption for a long time with spoofed static IPv6 addresses. By changing the node's IPv6 address periodically, *ADW* can be reduced. Hence, reducing the attacker's disruption window and prohibits the attacker node to gain sufficient meaningful information for further disruptions. This study uses private addressing scheme of [28] and its extended version in [25] to ascertain, that by the use of temporary addresses, the attacker's disruption window can be reduced. During *ADW*, a successful spoofing attempt leaves significant disturbances in the network. One such damage as shown in [24], is corrupting the routing table of the border router. In this work, we show that by using the private-temporary addresses, the corrupted routing table can be repaired thus, exhibiting self-healing property in 6LoWPAN.

The main contributions are:

1. *Time-To-Live* and *Attack Disruption Window* parameters are incorporated in the attack tree of [24] to find temporal characteristics of the spoofing attack.
2. Simulation is performed to repair the corrupted routing table of the border router using temporary - private IPv6 addressing of [28]. Thus, offering resilience against the disruption caused by IPv6 spoofing attack in a 6LoWPAN network.
3. The performance impact of a periodicity of IPv6 address change, on the communication cost, is compared for two private addressing schemes [28] and [39].

Rest of the sections are organised as follows: Section 2 describes related work. IPv6 spoofing threat along with *ADW* and *TTL* is described in Sections 3 and 4 describes privacy preserving IPv6 addressing scheme, Section 5 represents simulations and evaluation, and finally, Section 6 concludes the study.

2 Related work

6LoWPAN security is the primary concern in the applications like IoT, IP based wsn etc. where IPv6 is the obvious choice at the network layer [38]. 6LoWPAN is characterized by small packet size, low bandwidth and resource constrained nodes (battery life, storage, computing power) [22, 42]. Due to these factors, traditional security measures cannot be applied directly to it [27, 32]. Further, applications using 6LoWPAN protocol may render nodes to be exposed physically and therefore, can be compromised. Several crypto-based approaches are studied as summarized below in the literature, which can secure the network against the spoofing attack. However, due to the absence of fault-tolerant key management and resource intensive crypt-methods, there is a possibility of their failure.

MAC layer security provided by IEEE 802.15.4 standard may be sufficient, provided successful key management in place. As IEEE 802.15.4 standard does not provide any key management specification, and it is still an open research issue [12]. Authors in [21] have proposed a pairwise key establishment scheme for 6LoWPAN. However, if a node is compromised, then its pairwise keys can be used by a malicious node until the compromised node is detected and keys are revoked. Authors in [30] have proposed network access control to avoid communication with malicious nodes. Their proposed system increases each packet processing delay. Cryptographically Generated Address (CGA) [2, 17] provides security against spoofing attack as long as keys are not compromised and CGAs are certified. Authors in [35] have worked on the issue of false claim of ownership of IP address and proposed Crypto-ID using Elliptical Curve Cryptography (ECC), in IID part of the address, provisioning IP address authentication. Authors in [9] proposed a scheme with node authorization to obtain the IPv4 address in the MANET. Authors in [11] have developed security headers for 6LoWPAN networks. They have proposed compressed AH and ESP headers for 6LoWPAN which provides end-to-end security but have not discussed key management issues. Nodes authentication relies on pre-loading of keys in the nodes. Stealing of keys can make the crypto-solutions break [16]. Link- layer encryption and authentication may not be sufficient to provide confidentiality, authentication, integrity, to data as well as routing protocol packets [10]. Some works have been proposed to securely bootstrap the 6lowpan network based on mutual authentication between nodes and a base station [7, 19, 34, 43]. However, in [43] and [19], all the nodes need to pre-deploy with information for authentication and key generation which may be a bottleneck when network size increases. Scheme in [7] merges routing algorithm and authentication, which creates overheads for larger deployments. The scheme in [34] has higher communication overhead for performing authentication of nodes. Authors in [29] have used a scheme in which every IPv6 fragment is authenticated. Thus, the packet size is increased by 12%. Although, their scheme successfully secures fragments against the spoofing threat, the increased packet size results in higher energy consumption and packet processing delay. Authors in [40] have proposed wireless pre-loading scheme which uses physical layer key generation along with Diffie-Hellman key exchange. However, Diffie-Hellman's program memory consumption is high on many motes [23]. An experimental study in [15] has reported that in the Contiki operating system, link layer security used with phase optimization causes memory overhead. Authors in [20] have used the ECC based authentication of the mobile nodes with a diffie-Hellman key exchange in 6LoWPAN networks. However, the authentication process increases the handover delay and

their scheme is susceptible to Man-in-Middle attack. The scheme of [14] uses the two phase authentication for the device during the session establishment based on the past behavior collected from the legitimate device. If the malicious IoT device spoofs the identity of the legitimate device, its authentication fails due to mismatch between current behavior and the recorded behavior. IoT gateway puts the malicious device on a blacklist. However, the overhead at the IoT gateway increases as the authentication has to be performed for every session. The authors in [6] have used AES encryption for encrypting payload of application layer for IoT applications. Their scheme encrypts the AES key using Attribute Based Encryption(ABE). However, these two step encryption process before data transfer increases the transmission delay and consumes processing power.

Table 1 presents the security analysis of different category of techniques which is used to mitigate spoofing attack. These categories are Symmetric encryption based [6], Message authentication based [29], Cryptographically generated addresses [17], Device authentication based [14], temporary addresses(our scheme).

It is observed that there is a lack of approach which provides resistance against successful spoofing attempts by minimizing its disruption time and enables the network to self-heal. This study has demonstrated that the use of private-temporary addresses can reduce the attack disruption time. We have also shown that one such disruption can be corrected by the use of private addressing scheme proposed in [25, 28].

3 IPv6 address spoofing in 6LoWPANs

Routing Protocol for Low-Power and Lossy Networks (RPL) [41] is a dominant routing protocol for a route over routing in 6LoWPANs. RPL uses DODAG Information Solicitation (DIS), DODAG Information Object (DIO) and Destination Advertisement Object (DAO) messages for control operations regarding routing in the 6LoWPANs. Another associated protocol for 6LoWPAN is 6LoWPAN-Neighbor Discovery (6LoWPAN-ND) [36], which is used in a neighbor discovery process. 6LoWPAN-ND uses Neighbor Solicitation (NS) and Neighbor Advertisement (NA) control messages to perform the neighbor discovery. RPL lacks appropriate security implementations [1]. Various attacks like routing table overload, increased rank attack, DAG inconsistency attack, version number attack etc. are possible using RPL control messages [3, 26]. RPL control messages can be exploited to perform spoofing attack, eventually corrupting the routing table/neighbor cache [24]. A spoofing attack is an active internal or external attack, in which an attacker profess to be a victim to gain an illicit advantage [13]. IPv6 address can be spoofed and used to

Table 1 Security analysis comparison for spoofing attack mitigation techniques

Scheme	Mechanism	Security Property	Overhead	Address Privacy	Self-healing property
[6]	AES based encryption	Confidentiality, Integrity	Extra bytes in header, Resource overhead	Breaks in case of key compromise	No
[29]	Offline-Online SignCryption	Message authentication	Per fragment extra bytes, Resource overhead	No	No
[17]	CGA based	Confidentiality, Integrity	Certificate authority required	Breaks in case of key compromise	No
[14]	Fingerprint based	Device authentication	Large gateway overhead	No	No
Our scheme	Temporary addresses	Identity hiding	Minimal overhead at the routers	Yes	Yes

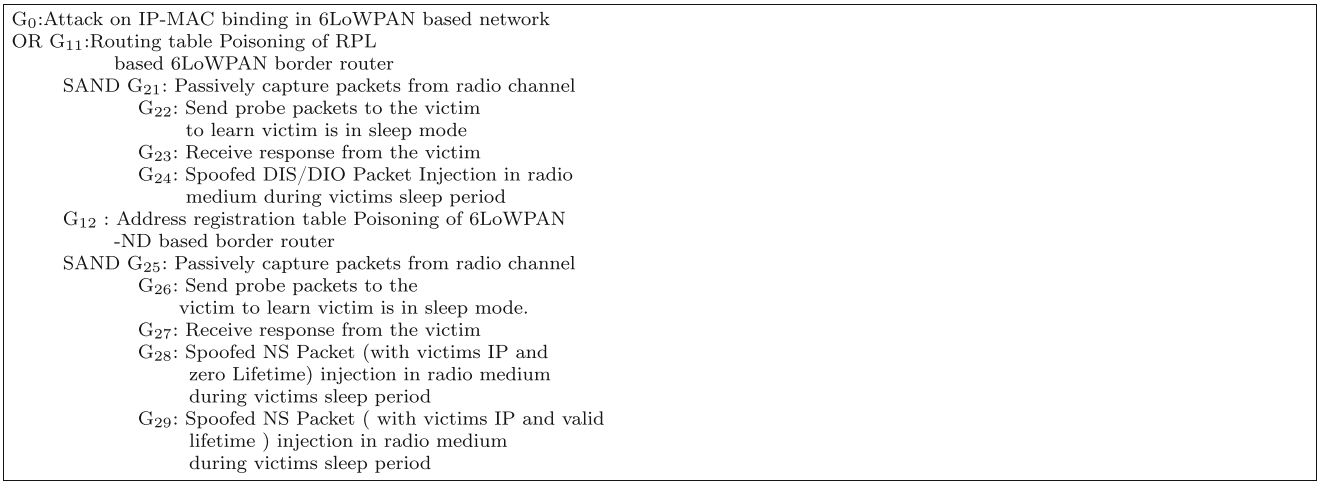


Fig. 1 Attack Tree [24]

launch further attacks. There are different ways in which spoofed address can be used. One such misuse of spoofed IPv6 address in the 6LoWPAN network is demonstrated in [24]. Authors have exploited spoofed RPL control messages to corrupt the routing table with wrong IP-MAC binding. They have used the attack tree modelling tool for analyses to find a probability of attack success and cost to the attacker. The corresponding attack tree is shown in Fig. 1.

However, this attack tree lacks the temporal analysis of the IPv6 spoofing attack. This study has incorporated the TTL parameter to the attack tree of Fig. 1 and eventually deriving the ADW. TTL gives time to perform micro attack at leaf nodes and eventually time to achieve the attack goal.

Subgoal G_{11} can be reached by the following path:

Intrusion scenario G_{11}: $G_{21} \otimes G_{22} \otimes G_{23} \otimes G_{24}$

Where $\otimes$ operator represents priority AND gate. TTL for the subgoal G_{11} is calculated as follows:

- $TTL_{G_{11}}$: To calculate the TTL for the subgoal G_{11}, formula used for AND gate is $\left\{ \sum_{G_i=1}^{n} TTL_{G_i} \right\}$ which is analogous to a formula used to find the cost in [24].

3.1 Time-To-Live (TTL)

TTL [5] implies the time during which an attacker should be able to finish the attack actions to achieve the attack goal eventually. By analyzing TTL, the total time to perform the spoofing attack can be derived. TTL values can be found by performing time complexity analyses of the attack tree of Fig. 1. The time complexity of the attack depends on the number of messages need to be transmitted and time to transmit them. Time complexity analysis is performed for subgoal G_{11} of Fig. 1.

3.1.1 Time complexity analysis

The time required for each micro event is approximated in this section. Table 2 shows analysis parameters used in this approximation.

Transmission time (t) to send one packet is given by Eq. 3.1.

$$t = T_{onair} + T_{csma} \tag{3.1}$$

Where, T_{onair} is given by $T_{onair} = (T_{tx} + T_{pr})$. Assuming data rate of 250 kbps, $T_{tx} = \frac{packetSize}{250kbps}$, radio propagation time per byte is calculated as $T_{pr} = \frac{d}{3\times10^8\ m/s} \times 8$, leading to value of T_{onair} as given by Eq. 3.2.

$$T_{onair} = \left(32 \times 10^{-6} + 2.6 \times 10^{-8} \times d\right) \times packetsize \tag{3.2}$$

Where d represents a distance between the attacker and the victim.

Assuming *cc2420* transceiver which uses CSMA/CA algorithm for radio channel access, T_{csma} is given by Eq. 3.3.

$$T_{csma} = (C_t + B_t) \times NB \tag{3.3}$$

where time to perform CCA (C_t) takes 8 symbol period or 128 µs, Number of *back off* slots *(NB)* ranges from 0 to 4 (default is 3). One *back off* period (B_t) takes 20 symbol period, symbol rate of 2.4 GHz- IEEE 802.15.4 standard is 62.5 KSymbols/second. Therefore, one back-off period is as $\frac{20}{62500} = 320\ \mu s$ and number of back off periods is in the interval is $\left[0 : 2^{BE} - 1\right]$ where BE ranges from 3 to 8 (default is 4) [18]. Hence, rewriting Eq. 3.3 and given by

$$T_{csma} = \left(125\ \mu s + \left(2^{BE} - 1\right) \times 320\ \mu s\right) \times NB \tag{3.4}$$

Table 2 Analysis parameters

Symbols	Meaning
TTL	Total time to perform root goal G_{11} of Fig. 1
TTL_{21}	Time-To-Live for micro event G_{21} of Fig. 1
TTL_{22}	Time-To-Live for micro event G_{22} of Fig. 1
TTL_{23}	Time-To-Live for micro event G_{23} of Fig. 1
TTL_{24}	Time-To-Live for micro event G_{24} of Fig. 1
t_{dio}	Time to send one DIO packet
t_{dis}	Time to send one DIS packet
t_{echorq}	Time to send one Echo Request packet
t_{echors}	Time to send one Echo Response packet
T_{onair}	Time for message to reach the destination i.e time the message travels in air
T_{csma}	Time needed to perform carrier sense multiple access (CSMA) operation
T_{tx}	Transmission time of message from node to wireless link
T_{pr}	Radio signal propagation time
s_{dio}	Size of DIO packet
s_{dis}	Size of DIS packet
s_{echorq}	Size of Echo Request packet
s_{echors}	Size of Echo Response packet
c_t	Time required for Clear Channel Assessment (CCA)
NB	Number of Back off slots in CSMA/CA operation
B_t	Back off period
BE	Back off exponent

Upper bound on T_{csma} can be found by putting maximum values of BE and NB i.e 8 and 4 respectively.

t_{min} is the time required to send one packet where a node gets the access of the medium after performing CCA. In this case, T_{csma} is equal to the time to perform CCA (C_t). Thus, t_{min} is given by Eq. 3.5

$$t_{min} = T_{onair} + C_t = \left(32 \times 10^{-6} + 2.6 \times 10^{-8} \times d\right) \times packetsize + 125\ \mu s \quad (3.5)$$

t_{max} is the maximum time node takes to transmit the packet in the presence of simultaneous traffic in the channel. Thus, t_{max} is derived from Eqs. 3.1, 3.2 and 3.4 and given in Eq. 3.6.

$$t_{max} = \left(32 \times 10^{-6} + 2.6 \times 10^{-8} \times d\right) \times packetsize + \left(125\ \mu s + \left(2^{BE} - 1\right) \times 320\ \mu s\right) \times NB \quad (3.6)$$

1. TTL for G_{21}, i.e. receiving DIO message:
   Here we assume that one DIO message is sufficient to learn victim's IPv6 address. Thus, time to perform event G_{21} is time to receive one DIO message which is given as,
   $$TTL_{21} = T_{tx} = \frac{s_{dio}}{250kbps} \quad (3.7)$$
   Where s_{dio} is 141 bytes (25 bytes MAC header, 6 bytes physical header and 110 bytes DIO header - a total of 141 bytes) [18, 41].
2. TTL for G_{22}, i.e. sending Echo-Request message:
   The minimum time to send one echo-request packet, is t_{min} and The maximum time is t_{max} and the *packetSize* (s_{echorq}) is 73 bytes (25 bytes MAC header, 6 bytes physical header and 42 bytes of ICMPv6 Echo Request header). We set a number of echo-request `(noEchorq)` to be sent as 5 in order to counter for losses in the network. Thus, TTL_{min} and TTL_{max} is given by:
   $$minTTL_{22} = (noEchorq \times t_{min}) + turnTime \quad (3.8)$$
   $$maxTTL_{22} = (noEchorq \times t_{max}) + turnTime \quad (3.9)$$
   Where the *turnTime* is Rx to Tx switching time of the node.
3. TTL for G_{23}, i.e. waiting for Echo-Response message
   Attacker's waiting time for this event has to consider turnTime of victim node and time to send echo response packets and given by Eqs. 3.10 and 3.11
   $$minTTL_{23} = (noEchors \times t_{min}) + turnTime \quad (3.10)$$
   $$maxTTL_{23} = (noEchors \times t_{max}) + turnTime \quad (3.11)$$
   Where the *packetSize* (s_{echors}) is 73 bytes (25 bytes MAC header, 6 bytes physical header and 42 bytes of ICMPv6 Echo Response header).

4. TTL for G_{24}, i.e. transmitting spoofed DIO message
   `noDio` is the number of DIO messages in order to counter for losses in the network. Thus, TTL_{min} and TTL_{max} is given by:

$$minTTL_{24} = (noDio \times t_{min}) + turnTime \quad (3.12)$$

$$maxTTL_{24} = (noDio \times t_{max}) + turnTime \quad (3.13)$$

   Where the *packetSize* is s_{dio}.

All the TTL values are affected if the channel is prone to transmission errors. To model errors in the timing analysis, the probability of successful packet transmission under errors is considered. Let E be the bit error rate, P_p be the successful packet transmission under the error condition and given by Eq. 3.14,

$$P_p = (1-E)^{packetSize} \simeq 1 - (E \times packetSize) \quad (3.14)$$

Probability of successful trial after nth trial is $(1-P_p)^{n-1} \times P_p$.

Mean number of attempts out of n trials before successful transmission is

$$\sum_{n \geq 1} n \times (1-P_p)^{n-1} \times P_p \simeq \frac{1}{P_p} \quad (3.15)$$

Therefore, from Eq. 3.15, the average time required to transmit a packet successfully is given by

$$t_{avg} = \frac{t}{P_p} \quad (3.16)$$

Thus, averaging Eqs. 3.7–3.13 under error conditions, minimum and maximum average TTL values are $TTL_{avg} = \frac{TTL}{P_p}$ i.e $TTL_{avg22} = \frac{TTL}{P_{p22}}$, $TTL_{avg23} = \frac{TTL}{P_{p23}}$, $TTL_{avg24} = \frac{TTL}{P_{p24}}$.

Lower bound and upper bound to perform attack using subgoal 1(G_{11}) is given by Eqs. 3.17 and 3.18.

$$minTotalTTL = TTL_{21} + \frac{minTTL_{22}}{P_{p22}} + \frac{minTTL_{23}}{P_{p23}} + \frac{minTTL_{24}}{P_{p24}} \quad (3.17)$$

$$maxTotalTTL = TTL_{21} + \frac{maxTTL_{22}}{P_{p22}} + \frac{maxTTL_{23}}{P_{p23}} + \frac{maxTTL_{24}}{P_{p24}} \quad (3.18)$$

Back off exponent (BE) and a number of back-off periods (NB) affect the maximum TTL. Figure 2 shows the effect of BE and NB on the upper bound on TTL value. It is shown that higher back off periods with a higher *back off* exponents increases the maximum TTL for the attacker. BE and NB are increased if collisions are experienced.

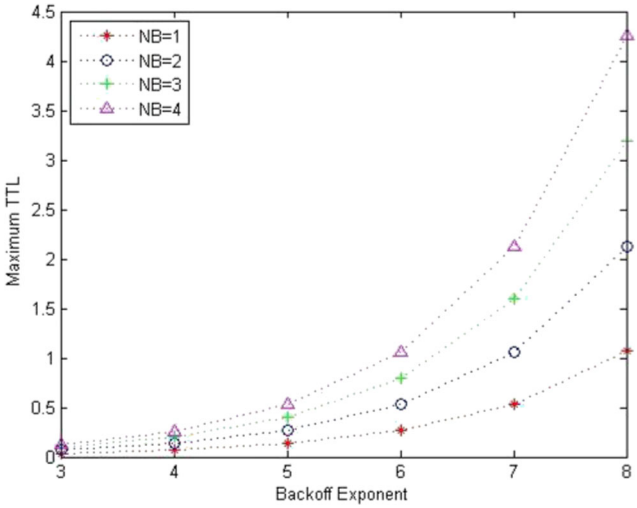


Fig. 2 Maximum TTL

Therefore, there could be simultaneous transmissions due to high congestion. It causes the higher values of BE and NB. Thus, the maximum TTL value is increased.

Figures 3 and 4 show the effects of varying BE and NB on each micro goal. It is observed that attack micro goal 2 and 3 contribute maximum in overall TTL value.

3.2 Attack disruption window

Definition: `Attack Disruption Window (ADW)` is defined as the time during which successful spoofing attempt can disrupt the network.

When an attacker node successfully spoofs the IPv6 address of the victim node, it can use this address to launch DoS, DDoS, flooding attacks, Man-In-The-Middle attack etc. in the network. Even attacker can cause harm to victim node by launching Man-In-the Middle attack, packet

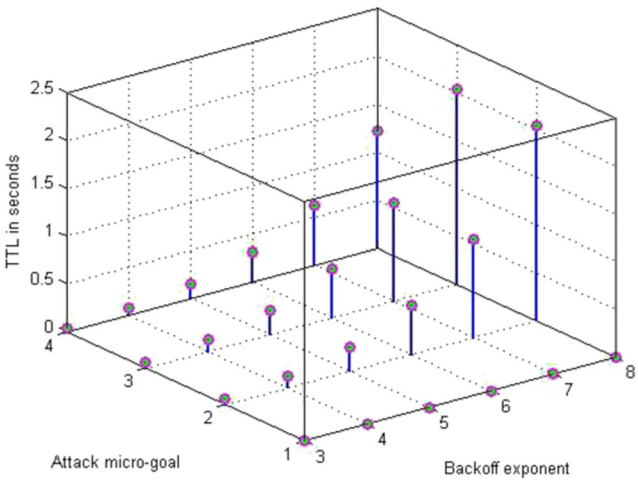


Fig. 3 Goal wise TTL with varying Backoff exponent

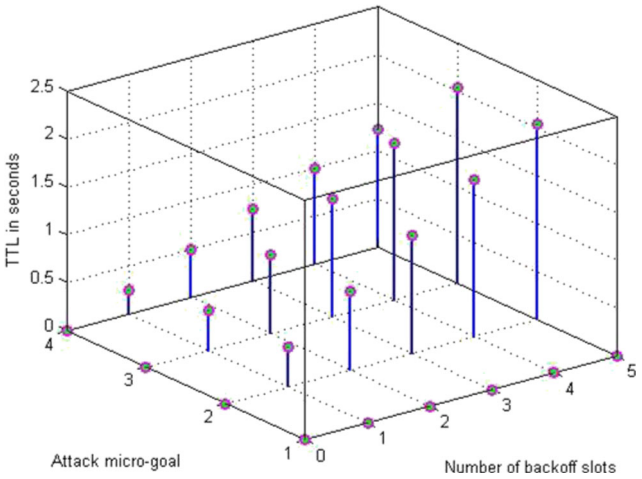


Fig. 4 Goal wise TTL with varying number of back-off slots

modification attack or monitors victim's activities to get sensitive information by learning and tracking the victim's IPv6 address. It can even prohibit victim to form neighbor relationships. On the failure of spoofing defence measures in the network, spoofing attempt can cause damage to the network until the time threat is detected and corrected. With the advent of IoT, all tiny but smart devices also join the network. Due to resource constraints, cryptographic solutions may not be a successful solution to save the IoT networks from spoofing threats. Anyone sneaking into the unsecured wireless channel can gain access to node's IPv6 addresses and performs further attacks. Therefore, it is necessary to make network resilient against spoofing threats, minimizing the ADW length. ADW is given by Eq. 3.19 where *totaltime* is the time till which spoofing attack is detected and corrected.

$$ADW = totaltime - TTL \tag{3.19}$$

To make network resilient against spoofing attacks, it is necessary to decrease the length of ADW. This study proposes to deprecate IPv6 address of the node and assign new addresses periodically. If IPv6 address changes periodically, then the spoofed address becomes invalid for further communications in the network. Thus, the length of ADW is till the time current IPv6 address of the victim is active. Once, it is deprecated; an attacker will not be able to cause disruption to the victim node using the spoofed address. So the Eq. 3.19 now becomes

$$ADW = currentaddress\_activetime - TTL \tag{3.20}$$

Value of TTL is highly dependent on CSMA parameters NB and BE as shown in the Fig. 2. Lower values of these parameters indicate congestion less traffic scenario in the network. In that case, ADW becomes approximately equal to the active time of current address. As *currentaddress_activetime* is reduced, ADW is also reduced. Lesser the *currentaddress_activetime,* the frequent address changes are required. However, the management of temporary addresses creates communication overhead. Therefore, it is the trade-off between addresses change frequency and length of `ADW`. As periodicity of address change is increased, `ADW` length decreases. Figure 5 shows an effect of address change periodicity on the `ADW` length. It is observed that `ADW` length increase linearly with the increase in address change periodicity.

This study evaluates the communication cost of privacy addressing scheme for 6LoWPAN [28] for minimizing ADW length.

4 Resiliency towards IPv6 spoofing attack

An attacker is successful in binding its own MAC address to the IPv6 address of the victim as shown in [24]. Due to this, victim node is not able to register itself to the router, leading to a denial of service because of corrupted routing table/neighbor cache as shown in Fig. 6. In the example network of 5 nodes, N1 acts as a border router, N2 is victim node, N4 is a malicious insider node. N4 can successfully spoof the N2's IPv6 address and have a corresponding entry in the neighbour cache of the N1. N2 is not able to have an entry in the cache as N4 spoofs its IPv6 address, and it is already registered with the router before N2 could register. Now, N2 is not able to register with the router and causing denial of service. At the same time, N2's traffic is going to N4, resulting into confidentiality breach.

As discussed in the previous section, when temporary addresses are used, the victim assumes a new IPv6 address and deprecates the old address. This allows a victim to register itself with the router. Now, the victim's IPv6 address

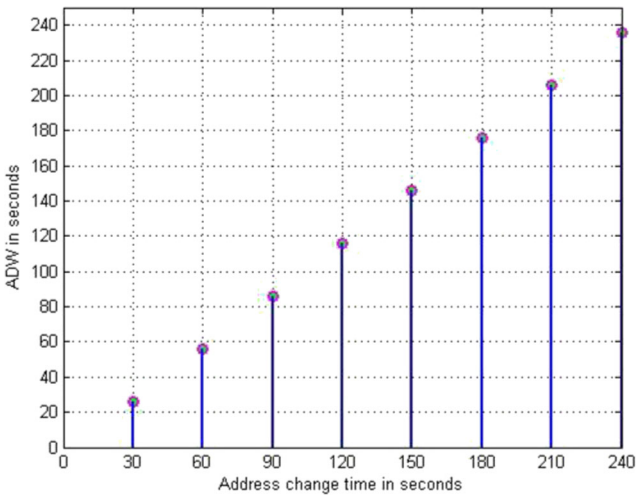


Fig. 5 ADW vs Address change periodicity

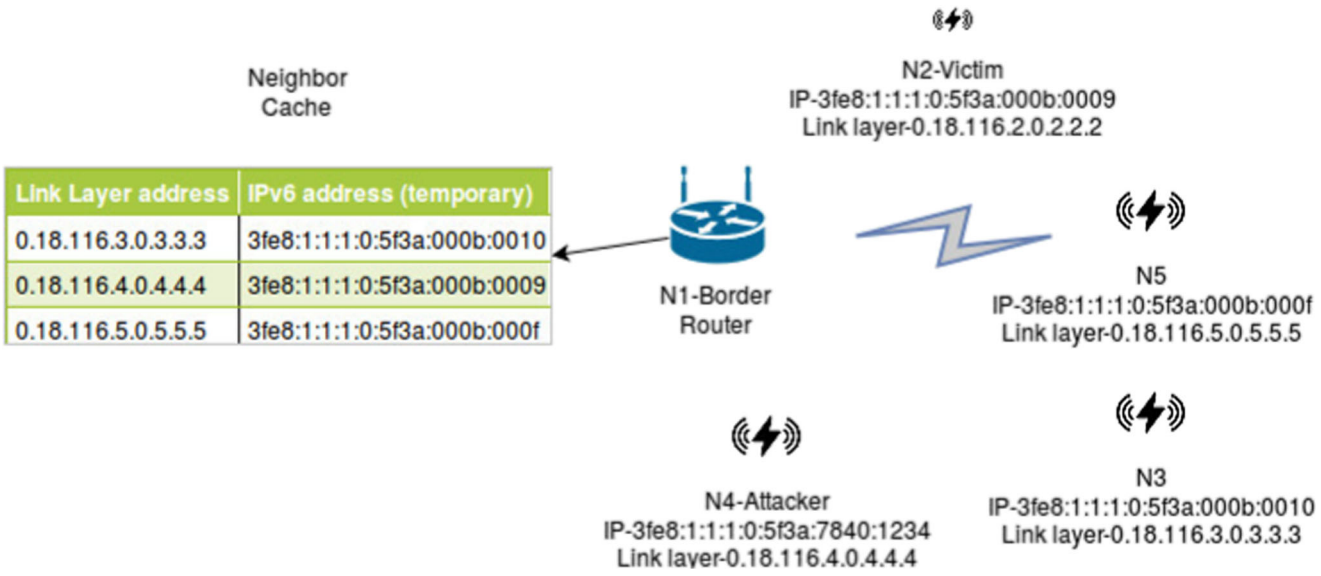


Fig. 6 Threat Model

is bound to its own MAC address and attacker's MAC address is bound to the victim's old address which is deprecated. Hence, the routing table/neighbor cache is being repaired as shown in Fig. 7. The address which attacker has spoofed is no longer valid thus reducing the disruption in the network.

We have used privacy addressing scheme in [28] as a measure to provide resistance to the spoofing attack. Application of its extended version as proposed in [25] ensures MAC address changes along with IPv6 address change, thus enabling resilience against MAC spoofing as well. A scheme in [28] assumes network to be divided into interconnection of single-hop 6LoWPANs, in which multi-hop connectivity is possible via routers of each 6LoWPAN. Every single 6LoWPAN is connected to the IPv6 Internet via an edge router. In this scheme, every node generates a set of IPv6 addresses from the minimal information received from the router. IPv6 address is divided into two parts: global routable IPv6 address and Interface Identification

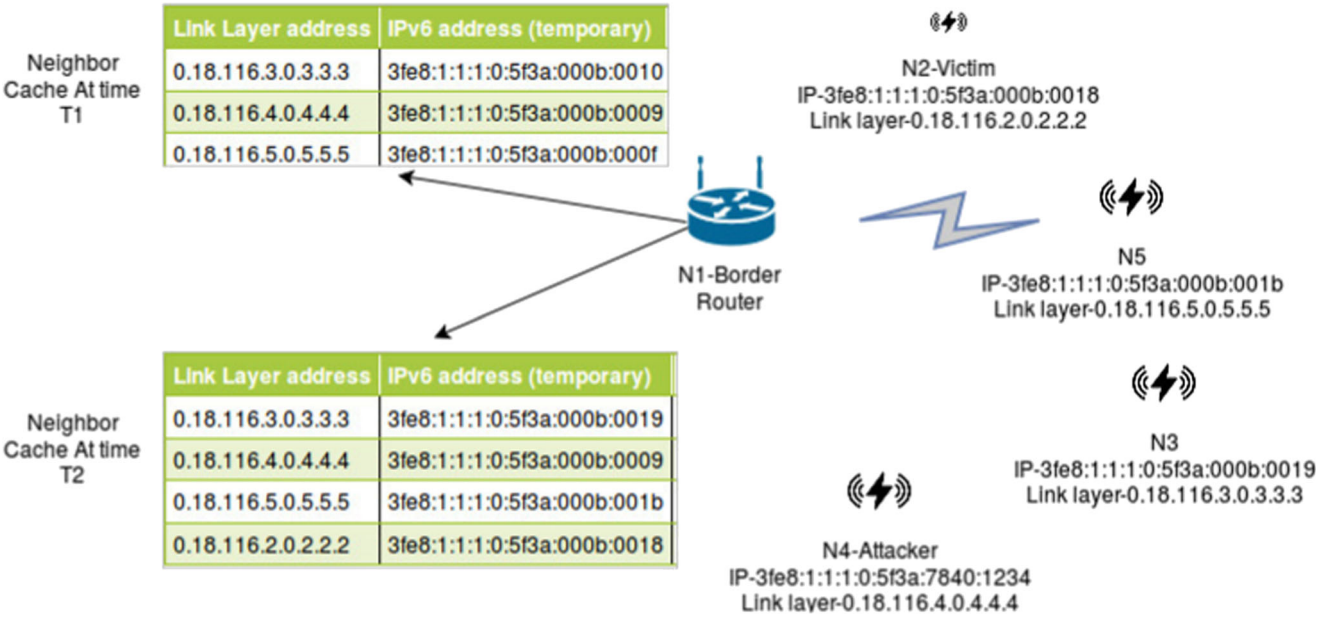


Fig. 7 Resiliency against the threat model

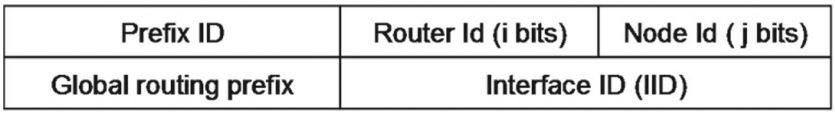


Fig. 8 IPv6 address structure proposed in [28]

(IID). IID part is divided into `Node Id` and `Router Id` as shown in Fig. 8.

In a single-hop 6LoWPAN, nodes receive congruence seeds from the router and generate `Node Id` and `Router Id` parts of the address. Every node is given a unique id (UID - `node_UID` and `router_UID`) during the deployment phase. Each node formulates a set of unique and non-repeatable number sequence using congruence class of its UID and congruence seed received from the router. Two different congruence seeds are used for `Node Id` and `Router Id` parts in IID. Each node registers its `node_UID` with the router in network initialization phase. Once, a router receives all the `node_UID`s; it sends congruence seed (CS_R) which is randomly generated, and at least two greater than the highest received `node_UID` s. At the same time, the router also assumes dummy `node_UID` which is one greater than the highest received `node_UID` s. Similarly, all the routers register their respective `router_UID` s with the edge router. Once, an edge router receives all the `router_UID` s; it generates a congruence seed (CS_{ER}) which is greater than the highest received `router_UID`s. Now each node generates `Node Id` from the sequence generated from congruence class $[node\_ID]_{CS_R}$ and `Router Id` from $[router\_ID]_{CS_{ER}}$. These sequences are unique for each node, and therefore, addresses generated are unique, avoiding Duplicate Address Detection (DAD) process. Nodes can generate pseudo-random and unique address sets, which can be changed periodically.

Taking an example: a 6LoWPAN consists of a router with `router_UID` = 12, and 2 nodes with `node_UIDs` 13 and 25. Let the global routing prefix is *6786::76DE*. The CS_{ER} is 35 (higher than `router_UID` 12 in the subnet). The CS_R = 30 (greater than highest *UID* of nodes in the PAN, i.e. 25).

Address calculation for node 13 in 6LoWPAN1:

CS_R (congruence seed generated by a router) = 30 (greater than highest *UID* of nodes in the PAN, i.e. 25)

Congruence class for `Node Id` part of node with *UID* 13 = $[\texttt{node\_UID}]_{CS_R} = [13]_{30} = \{13, 43, 73, \ldots\}$

Congruence class for `Router Id` part of node with *UID* 13 = $[\texttt{router\_UID}]_{CS_{ER}} = [12]_{35} = \{12, 47, 82, \ldots\}$

At time *T1*,

`Node Id` is chosen randomly from sequence of $[13]_{30}$ and `Router Id` is chosen randomly from the sequence of $[12]_{35}$, values are:

`Node Id` = 43 i.e *0x002B*
`Router Id` = 47 i.e *0x002F*

Combining global routing prefix, `Node Id` and `RouterId`, the IPv6 address for the node_UID 13 is *6786::76DE:002F:002B*.

At time *T2*, choosing another number randomly from $[13]_{30}$ and $[12]_{35}$ address generated is *6786::76DE* :0075:0085.

The periodicity of address change can reduce the spoofing attack impact by reducing the attack disruption window. Enhancing security of the network by the use of temporary addresses is a vital solution as it allows a network to self-heal from the disruptions already caused by the successful spoofing attempt. Frequent change of address can reduce ADW length, but it comes with a cost of communication overhead. However, an optimum value for a frequency change can be used to get the least communication cost. This study has simulated the communication cost using this scheme with different periodicity.

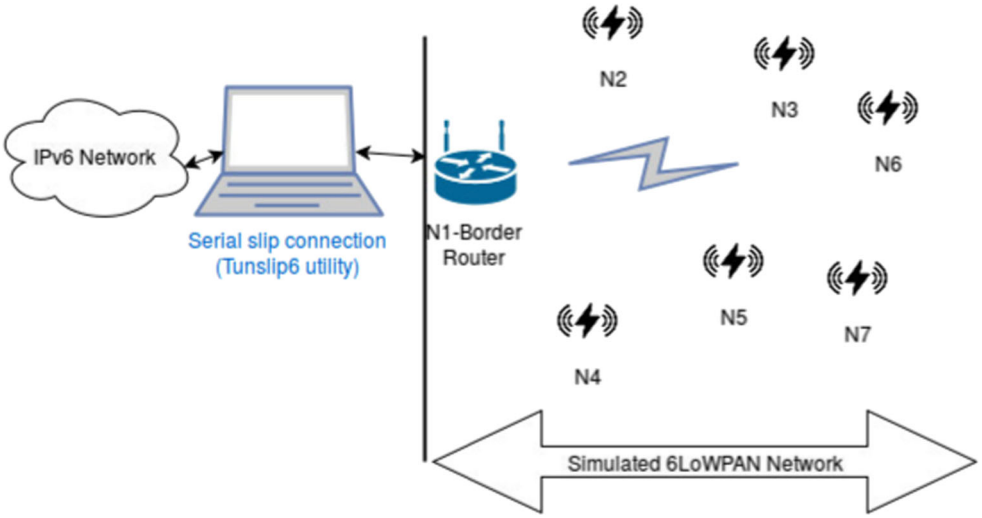
5 Simulation

An attack is performed in the simulated network using Contiki operating system's [8] Cooja [31] simulator. In a simulation, nodes are deployed in a random manner using 100 m × 100 m area. Each node defers for a small amount of time before starting the communication to avoid the collisions.

5.1 Simulation set up

Simulated network is shown in Fig. 9, which is a single hop network. The choice of a single hop network is because the attacker tries to spoof the nodes IPv6 addresses, which come in its radio range by listening to radio messages, i.e. local attacker. However, the remotely initiated attack can also be mitigated as solution implemented is host initiated. Each node in the network becomes the part of DODAG, whose root is node 1. Each node has at least one global routable address whose prefix is disseminated by RPL border router running on node 1. These are the permanent addresses for the nodes. Global prefix for the

Fig. 9 Simulated Network



permanent address is given by tunslip6 [37] utility, running outside Cooja, on host operating system which is Ubuntu 14.04. To demonstrate the resilience against the spoofing attempts using the private temporary addresses, few sets of nodes run addressing mechanism as described in [28] protocol to generate the set of temporary addresses. Thus, in addition to the permanent address, few sets of nodes have temporary addresses which are used in communication and changed periodically. Privacy addressing scheme requires one edge router to disseminate global routable prefix and router in every single 6LoWPAN to disseminate address configuration parameters (prefix and random seeds). Thus, the simulated network consists of 7 nodes, where node 1 runs Contiki's border router (modified to print layer two addresses along with IP addresses of the neighbours) which disseminate prefix for a global routable permanent address. Node 1 is also a root of DODAG. Node 2 runs edge router needed to run addressing protocol of [28] which disseminates global routable prefix for the temporary addresses. Node 3 runs router as described in [28], which disseminates remaining address configuration parameters. Rest of the nodes are normal nodes and can be victim nodes. Node 7 runs attacker code which sends spoofed RPL control messages.

5.1.1 Changes made in contiki's core

Scheme [28] is implemented using rime communication driver, and RPL needs a sicslopan driver of Contiki. To show resiliency against spoofing attack using privacy addressing scheme, we have made few changes to Contiki's core. We have defined two communication drivers in netstack.h header file of Contiki. Two drivers are NETSTACK_CONF_NETWORK1 and NETSTACK_CONF_NETWORK2 each for sicslowpan_driver and rime_driver respectively and enabled both of them in configuration files. While sending packets, contiki uses appropriate drivers to send rime packets and RPL packets respectively. However, while receiving, all the packets go to the first network driver defined in MAC or security driver. To overcome this issue, we have used one reserved bit in IEEE 802.15.4 MAC header's frame control field (contiki uses IEEE 802.15.4 framer) as a flag. When it is 0, an incoming packet is given to sicslopan_driver otherwise incoming packet is given to rime_driver. This flag is set at the sender side by rime_driver while sending rime packets. It is reset by the sicslopan_driver while sending RPL packets.

5.1.2 Simulation parameters

Simulator parameter for cooja are shown in Table 3 and for Contiki, it is shown in Table 4. For the scheme in Table [28], 8 bits are allocated for the `NodeId` part and the `Router Id` part each.

Table 3 Cooja simulator parameters

Radio model parameters	Value
Antenna	Omni directional
Radio model	UDGM- Distance loss
Transmitter output power (dBm)	0
Receiver sensitivity (dBm)	−94
Radio frequency	2.4GHz
BE, NB	3

Table 4 Contiki parameters

Parameters	Value
NETSTACK_CONF_NETWORK1	sicslowpan_driver
NETSTACK_CONF_NETWORK2	rime_driver
NETSTACK_CONF_WITH_IPV6	1
NETSTACK_CONF_WITH_RIME	1
UIP_CONF_IPV6_RPL	1
NETSTACK_CONF_LLSEC	nullsec_driver
NETSTACK_CONF_MAC	csma_driver
NETSTACK_CONF_RDC	nullrdc_driver
NETSTACK_CONF_RADIO	cc2420_driver
NETSTACK_CONF_FRAMER	framer_802154

5.2 Evaluation and Results

Two sets of experiments are run:

- To show that through the use of private addresses, the corrupted routing table can be repaired, exhibiting self-healing nature.
- Effect of address change periodicity on communication cost to the nodes.

5.2.1 Routing table repair

Table 5 shows IPv6 addresses associated with each node.

Snapshot of a corrupted routing table entry of Contiki's RPL Border-Router is shown in the Table 6. Attacker node is having IPv6 address as `aaaa::212:7407:7:707` (permanent) and layer 2 address as `0.18.116.7.0.7.7.7.` Victim node is having IPv6 address as `aaaa::212:7404:4:404` (permanent), 3fe8:1:1:1:0:5f3a:000b:000f (temporary) and layer 2 address as `0.18.116.4.0.4.4.4.`. However, as shown in the second entry of a corrupted routing table, an attacker is successful in binding its link-layer address with victim's temporary IPv6 address.

From the Table 6, it is evident that the attacker node has successfully spoofed the victim node's IPv6 address. Node 4 does not have an entry in the table thus all the traffic belonging to node 4 is going to node 7 (attacker node). Thus, causing a DoS attack on the victim and eventually confidentiality breach to node 4. Now, due to the use of temporary addresses in the communication, nodes 4, 5, 6 changes their addresses after 60 s. Snapshot of a repaired routing table entry of Contiki's RPL Border-Router, taken after 60 s, is shown in the Table 7. Node 4's new address is 3fe8:1:1:1:0:5f3a:0013:001a and thus

Table 5 Nodes and their IPv6 addresses

Node id	IPv6 address with Link local scope	IPv6 address with global scope (permanent)	IPv6 address with global scope (temporary -private) - first two addresses from the set	Rime address
1	`fe80::212:7401:1:101`	`aaaa::212:7401:1:101`	-	0.18.116.1.0.1.1.1
2	`fe80::212:7402:2:202`	`aaaa::212:7402:2:202`	-	0.18.116.2.0.2.2.2
3	`fe80::212:7403:3:303`	`aaaa::212:7403:3:303`	-	0.18.116.3.0.3.3.3
4	`fe80::212:7404:4:404`	`aaaa::212:7404:4:404`	3fe8:1:1:1:0:5f3a:000b:000f, 3fe8:1:1:1:0:5f3a:000b:001a	0.18.116.4.0.4.4.4
5	`fe80::212:7405:5:505`	`aaaa::212:7405:5:505`	3fe8:1:1:1:0:5f3a:000b:0010, 3fe8:1:1:1:0:5f3a:000b:001b	0.18.116.5.0.5.5.5
6	`fe80::212:7406:6:606`	`aaaa::212:7406:6:606`	3fe8:1:1:1:0:5f3a:000b:0011, 3fe8:1:1:1:0:5f3a:000b:001c	0.18.116.6.0.6.6.6
7	`fe80::212:7407:7:707`	`aaaa::212:7407:7:707`	Any spoofed address	0.18.116.7.0.7.7.7

Table 6 Corrupted routing table

Link layer address	IPv6 address (temporary)
0.18.116.5.0.5.5.5	3fe8:1:1:1:0:5f3a:000b:0010
0.18.116.7.0.7.7.7	3fe8:1:1:1:0:5f3a:000b:000f
0.18.116.6.0.6.6.6	3fe8:1:1:1:0:5f3a:000b:0011
0.18.116.2.0.2.2.2	-

has an entry in the routing table. Node 7's entry is still using node 4's old IPv6 address which is now invalid. Thus, all the communication to node 4 uses its new address and packets are correctly delivered to node 4. Spoofed address by the attacker node (3fe8:1:1:1:0:5f3a:000b:000f) is no longer valid. To become effective again, the attacker has to spoof the new address again. However, the severity of an attack is again limited to next address change, limiting ADW. This shows that through the use of private-temporary addresses, the network can be made resilient against the spoofing attack by reducing ADW.

To ensure resiliency against layer 2 address spoofing, extended version of [28] as proposed in [25] is used to change layer 2 address of nodes along with IPv6 address. In that case, repaired routing table becomes as shown in Table 8.

5.2.2 Periodicity of address change vs communication cost

ADW can be reduced if nodes change their addresses frequent enough. Thus, it gives less time to an attacker to launch spoofing related attacks, and even if it is successful in creating some disruption, its effect is reduced, due to periodically new identities assumed by the nodes. However, frequent address change incurs the communication cost which is a prominent factor for energy consumption in low power and lossy networks. Figure 10 shows the number of packets observed in the 12-node network with one node as an edge router to give global routable prefix and one node as a PAN router. As the frequency of address change increases, the number of packets decreases. For comparison, two different privacy enabled addressing schemes are taken. The first scheme chosen in this study [28] is compared with the second scheme in [39]. The second scheme uses a cluster-based approach in which the cluster head of each cluster is responsible for address set allocation and maintenance. Nodes ask for new address set to cluster head after they are exhausted with the currently allocated address set. Address set consists of continuous addresses which can be randomly chosen by the nodes. Using both the schemes, series of experiments are run with a different periodicity of address change with each simulation running ten times with different random seed each time and observed average communication cost is shown in the Fig. 10a and b. The first scheme results in better communication cost compared to the second scheme. Results show that address change frequency beyond 120 s shows negligible variation in communication cost for the first scheme whereas, for the second scheme, some variation is observed.

Table 7 Repaired routing table

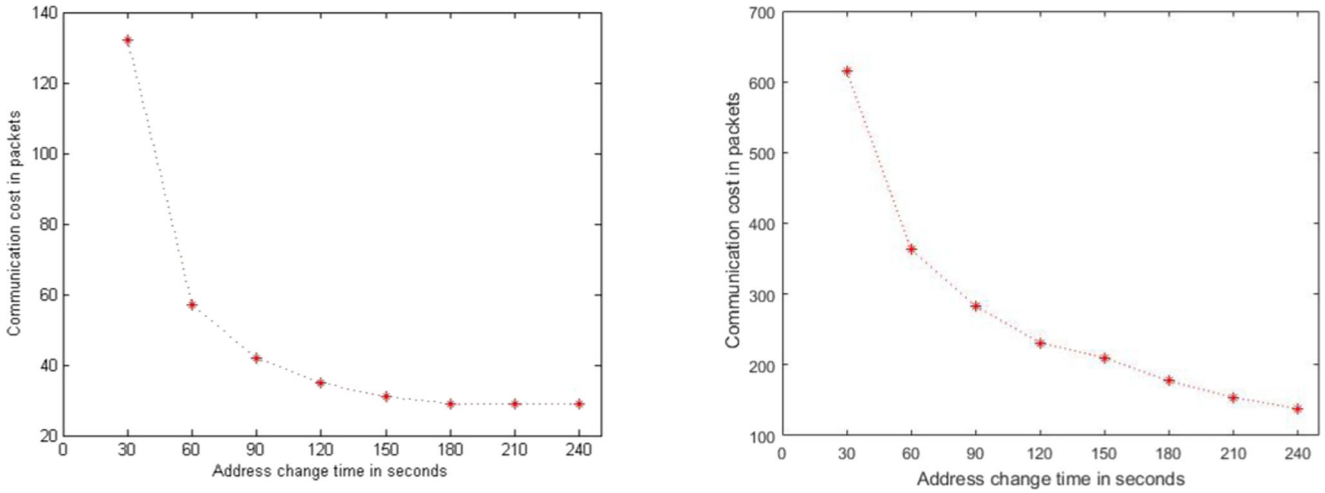
Link layer address	IPv6 address (temporary)
0.18.116.5.0.5.5.5	3fe8:1:1:1:0:5f3a:0013:001b
0.18.116.7.0.7.7.7	3fe8:1:1:1:0:5f3a:000b:000f
0.18.116.6.0.6.6.6	3fe8:1:1:1:0:5f3a:0013:001c
0.18.116.2.0.2.2.2	-
0.18.116.4.0.4.4.4	3fe8:1:1:1:0:5f3a:0013:001a

Table 8 Repaired routing table with link layer address change

Link layer address	IPv6 address (temporary)
1.0.5f.3a.00.13.00.1b	3fe8:1:1:1:0:5f3a:0013:001b
0.18.116.7.0.7.7.7	3fe8:1:1:1:0:5f3a:000b:000f
1.0.5f.3a.00.13.00.1c	3fe8:1:1:1:0:5f3a:0013:001c
0.18.116.2.0.2.2.2	-
1.0.5f.3a.00.13.00.1a	3fe8:1:1:1:0:5f3a:0013:001a

However, from the Fig. 5 address change frequency beyond 150 s, ADW increases linearly with increase in frequency. Address change frequency at 30 s gives minimum ADW as shown in the Fig. 5, but communication cost increases drastically. So optimum value of address change frequency could be 60 s to 90 s as there is a marginal difference in the communication cost between them.

Next set of experiments were performed to study the effect of address change on upper layer traffic. Most of the traffic in a resource-constrained network is connectionless to conserve the energy. Therefore, experimentation was done to evaluate the ability to intercept connectionless traffic correctly. 1000 test packets were sent in every 4 s and periodicity of address change was kept at 60 s resulting in 15 test packets between each address change. It was observed that for the 1000 test packets communication, the sender learns the new address of the receiver after two packets on an average in each 60 s interval. Further optimization can be done by decreasing the DIO message interval as the sender learns the new address from the incoming DIO messages. This way, a sender can learn new addresses quickly.



(a) For the scheme in [28]

(b) For the scheme in [39]

Fig. 10 Address change frequency vs communication cost

6 Conclusion and future work

This study has devised a solution to provide the resiliency in the presence of disruption caused due to spoofing threat. Attack disruption time depends on time to perform the attack and the address change periodicity. Time to perform the attack is derived by performing the time complexity analysis, and it is found that some CSMA parameters like back-off periods affect the time to perform the attack. An attacker can corrupt the routing table of the border router causing a denial of service to the victim using spoofed RPL messages. The corrupted routing table is repaired using periodically changing temporary-private addresses. Attack disruption time can be reduced with the increase in the periodicity of the address change. However, it increases the communication cost, which is the important parameter for energy consideration. An optimum value of address change periodicity for the optimum communication cost is observed to be 60 to 90 s in simulation. Thus, it is possible to resist disruptions caused by spoofing attempt by the use of private-temporary addresses and allows the network to self-heal.

References

1. Airehrour D, Gutierrez J, Ray SK (2016) Secure routing for internet of things: a survey. J Netw Comput Appl 66:198–213
2. Aura T (2005) Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard). http://www.ietf.org/rfc/rfc3972.txt. Updated by RFCs 4581, 4982
3. Badonnel AR, Mayzaud IC (2017) A distributed monitoring strategy for detecting version number attacks in rpl-based networks. IEEE Trans Netw Serv Manag 14(2):472–486. https://doi.org/10.1109/TNSM.2017.2705290
4. Barbir A, Murphy SL, Yang Y (2006) Generic Threats to Routing Protocols. Tech. Rep. 4593. https://doi.org/10.17487/RFC4593. https://rfc-editor.org/rfc/rfc4593.txt
5. Camtepe SA, Yener B (2007) Modeling and detection of complex attacks. In: 2007 Third international conference on security and privacy in communications networks and the workshops - securecomm 2007, pp 234–243. https://doi.org/10.1109/SECCOM.2007.4550338
6. Choi J, In Y, Park C, Seok S, Seo H, Kim H (2018) Secure iot framework and 2d architecture for end-to-end security. J Supercomput 74(8):3521–3535. https://doi.org/10.1007/s11227-016-1684-0
7. Chze PLR, Leong KS (2014) A secure multi-hop routing for iot communication. In: 2014 IEEE World forum on internet of things (WF-iot), pp 428–432. https://doi.org/10.1109/WF-IoT.2014.6803204
8. Dunkels A, Grönvall B, Voigt T (2004) Contiki - A lightweight and flexible operating system for tiny networked sensors. In: Proceedings - conference on local computer networks, LCN, pp 455–462. https://doi.org/10.1109/LCN.2004.38
9. Ghosh U, Datta R (2011) A secure dynamic ip configuration scheme for mobile ad hoc networks. Ad Hoc Netw 9(7):1327–1342. https://doi.org/10.1016/j.adhoc.2011.02.008
10. Gomez C, Kim E, Kaspar D, Bormann C (2012) Problem statement and requirements for IPv6 over low-power wireless personal area network (6LoWPAN) routing. RFC 6606, RFC Editor. https://tools.ietf.org/pdf/rfc6606.pdf
11. Granjal J, Monteiro E, Silva JS (2010) Enabling network-layer security on ipv6 wireless sensor networks. In: 2010 IEEE Global telecommunications conference GLOBECOM 2010, pp 1–6. https://doi.org/10.1109/GLOCOM.2010.5684293
12. Granjal J, Monteiro E, Silva JS (2015) Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun Surv Tutorials 17(3):1294–1312. https://doi.org/10.1109/COMST.2015.2388550
13. Granjal J, Monteiro E, Silva JS (2015) Security in the integration of low-power wireless sensor networks with the internet: a survey. Ad Hoc Netw 24:264–287
14. Gu T, Mohapatra P (2018) Bf-iot: Securing the iot networks via fingerprinting-based device authentication. In: 2018 IEEE 15Th international conference on mobile ad hoc and sensor systems (MASS), pp 254–262. https://doi.org/10.1109/MASS.2018.00047

15. Halcu I, Stamatescu G, Sgarciu V (2015) Enabling security on 6lowpan / ipv6 wireless sensor networks. In: 2015 7Th international conference on electronics, computers and artificial intelligence (ECAI), pp SSS–29–SSS–32. https://doi.org/10.1109/ECAI.2015.7301201
16. Hennebert C, Santos JD (2014) Security protocols and privacy issues into 6loWPAN stack: a synthesis. IEEE Internet J 1(5):384–398. https://doi.org/10.1109/JIOT.2014.2359538
17. Hossain M, Karim Y, Hasan R (2018) Secupan: a security scheme to mitigate fragmentation-based network attacks in 6lowpan. In: Proceedings of the eighth ACM conference on data and application security and privacy. ACM, pp 307–318
18. IEEE: Ieee 802.15.4 standard (2007) [Online] https://standards.ieee.org/about/get/802/802.15.html
19. Ikram M, Chowdhury AH, Zafar B, Cha HS, Kim K, Yoo SW, Kim D (2009) A simple lightweight authentic bootstrapping protocol for ipv6-based low rate wireless personal area networks (6lowpans). In: Proceedings of the 2009 international conference on wireless communications and mobile computing: connecting the world wirelessly, IWCMC '09. ACM, New York, pp 937–941. https://doi.org/10.1145/1582379.1582583
20. Jara AJ, Marin L, Skarmeta AF, Singh D, Bakul G, Kim D (2011) Mobility modeling and security validation of a mobility management scheme based on ecc for ip-based wireless sensor networks (6lowpan). In: 2011 Fifth international conference on innovative mobile and internet services in ubiquitous computing. IEEE, pp 491–496
21. Krentz KF, Rafiee H, Meinel C (2013) 6lowpan security: Adding compromise resilience to the 802.15.4 security sublayer. In: Proceedings of the international workshop on adaptive security, ASPI '13. ACM, New York, pp 1:1–1:10. https://doi.org/10.1145/2523501.2523502
22. Kushalnagar N, Montenegro G, Schumacher C (2007) Rfc 4919: Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals. IETF 31:45–75
23. Liu A, Ning P (2008) Tinyecc: a configurable library for elliptic curve cryptography in wireless sensor networks. In: Proceedings of the 7th international conference on information processing in sensor networks, IPSN '08. IEEE Computer Society, Washington, pp 245–256. https://doi.org/10.1109/IPSN.2008.47
24. Mavani M, Asawa K (2017) Modeling and analyses of ip spoofing attack in 6lowpan network. Comput Secur 70:95–110
25. Mavani M, Asawa K (2018) Privacy enabled disjoint and dynamic address auto-configuration protocol for 6lowpan. Ad Hoc Netw 79:72–86. https://doi.org/10.1016/j.adhoc.2018.06.010. http://www.sciencedirect.com/science/article/pii/S1570870518303627
26. Mayzaud A, Badonnel R, Chrisment I (2016) A taxonomy of attacks in rpl-based internet of things. Int J Netw Secur 18(3):459–473
27. Mishra A, Dixit A (2018) Resolving threats in iot: Id spoofing to ddos. In: 2018 9Th international conference on computing, communication and networking technologies (ICCCNT), pp 1–7. https://doi.org/10.1109/ICCCNT.2018.8493729
28. Mavani M, Asawa K (2017) Privacy preserving ipv6 address auto-configuration for internet of things. In: Intelligent communication and computational technologies. Springer, pp 577–584
29. Nikravan M, Movaghar A, Hosseinzadeh M (2019) A lightweight signcryption scheme for defense against fragment duplication attack in the 6lowpan networks. Peer-to-Peer Netw Appl 12(1):209–226. https://doi.org/10.1007/s12083-018-0659-8
30. Oliveira LML, Rodrigues JJPC, Neto C, De sousa AF (2013) Network admission control solution for 6LoWPAN networks. Proceedings - 7th international conference on innovative mobile and internet services in ubiquitous computing, IMIS 2013, pp 472–477. https://doi.org/10.1109/IMIS.2013.85
31. Osterlind F, Dunkels A, Eriksson J, Finne N, Voigt T (2006) Cross-level sensor network simulation with cooja. In: Proceedings 2006 31st IEEE conference on Local computer networks. IEEE, pp 641–648
32. Park S, Kim K, Haddad W, Chakrabarti S, Laganier J (2011) Ipv6 over low power wpan security analysis. IETF. ID draft-daniel-610wpan-security-analysis-05. Retrieved 10 May 2016
33. Pongle P, Chavan G (2015) A survey: attacks on rpl and 6lowpan in iot. In: 2015 International conference on pervasive computing (ICPC), pp 1–6. https://doi.org/10.1109/PERVASIVE.2015.7087034
34. Qiu Y, Ma M (2015) An authentication and key establishment scheme to enhance security for m2m in 6lowpans. In: 2015 IEEE International conference on communication workshop (ICCW), pp 2671–2676. https://doi.org/10.1109/ICCW.2015.7247582
35. Sarikaya B, Thubert P (2016) Address protected neighbor discovery for low-power and lossy networks. Internet-Draft draft-sarikaya-6lo-ap-nd-02, IETF Secretariat. http://www.ietf.org/internet-drafts/draft-sarikaya-6lo-ap-nd-02.txt
36. Shelby C, Nordmark B (2012) Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs). RFC 6775, RFC Editor. http://www.rfc-editor.org/rfc/rfc6775.txt
37. Simon DEA tunslip6 utility. https://github.com/contiki-os/contiki/blob/master/tools
38. Vasseur JP, Dunkels A (2010) Interconnecting smart objects with ip: The next internet. Morgan Kaufmann, San Mateo
39. Wang X, Mu Y (2015) Addressing and privacy support for 6lowpan. IEEE Sens J 15(9):5193–5201. https://doi.org/10.1109/JSEN.2015.2438002
40. Wilhelm M, Martinovic I, Uzun E, Schmitt JB (2010) Sudoku: Secure and usable deployment of keys on wireless sensors. In: 2010 6Th IEEE workshop on secure network protocols, pp 1–6. https://doi.org/10.1109/NPSEC.2010.5634458
41. Winter T, Brandt H (2012) RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, RFC Editor. http://www.rfc-editor.org/rfc/rfc6550.txt
42. Xiong K, Zhang Y, Zhang Z, Wang S, Zhong Z (2014) Pa-nemo: Proxy mobile ipv6-aided network mobility management scheme for 6lowpan. Elektron Elektrotechn 20(3):98–103
43. Yu H, He J (2012) Trust-based mutual authentication for bootstrapping in 6lowpan. JCM 7(8):634–642

Monali Mavani is a Research Scholar in Jaypee Institute of Information Technology, Noida, India. She was awarded Master in technology (CSE) in 2013 from Mumbai University, India and Bachlor in technology (EXTC) in 2002 from SNDT university, India. Her areas of interest and expertise are Wireless Networks, Network Security, Next Generation networks, Distributed Computing. Other than JIIT her employment association were also with SIES College, Mumbai India and Somaiya-Vidyavihar, Mubai, India. She is currently doing Phd in the area of 6LoWPAN security.

Krishna Asawa is working with the Jaypee Institute of Information Technology, Noida, India in the capacity of Professor. She was awarded Doctor of Philosophy (CSE) in 2002 from Banasthali Vidyapeeth University, India. Her areas of interest and expertise are Soft Computing and its Applications, Information Security, Knowledge and Data Engineering. Other than JIIT her employment association were also with National Institute of Technology, Jaipur, India and Banasthali Vidyapith, India.