A survey on security vulnerabilities in channel assignment algorithms of multi-interface wireless mesh networks

Maryam Asadzadeh Kaljahi and Mohsen Jahanshahi

ABSTRACT

Wireless mesh network (WMN) has been deployed as a key and progressing technology all around the world to provide the end-to-end users with high quality of service. Regarding of its major design concern as network throughput, WMN has been evolved based on multi-channel multi-radio strategy to make simultaneous transmissions possible. The advent of this significant improvement has introduced other challenging issues such as channel assignment (CA). Its core strategy applied in multi-radio multi-channel WMNs to promote achievable capacity using properly mapping the available channels to the radio interfaces. Although, in recent years, there are a number of research studies on CA’s improvement, there are some security drawbacks that should be addressed. Most of CA schemes assume all nodes involved in CA techniques are trustworthy. This unrealistic assumption and misbehavior of those nodes makes the network susceptible to various security attacks which lead to performance degradation. This paper presents an in-depth survey of the CA’s attacks as well as detection or prevention mechanism found in the literature. The attacks are classified into two groups: attacks imposed by the malicious neighbor as well as the assigner or the assignee. In addition, a comparative and analytical discussion is provided on current solutions and their open problems. According to conducted analysis, the current knowledge gap leading to most of these attacks is lack of accurate verification because of implicit trust in nodes in CA procedure. Although, there are some research studies proposing kinds of case-specific verification, they cannot close the gap properly, and there is a need to comprehensive approach. Finally, reviewing state-of-the-art of related works shows, there are more possible attacks such as collusion or multistage attacks that have not taken into account so far.

KEYWORDS

wireless mesh networks; multi-radio multi-channel; channel assignment attacks; secure channel assignment algorithms

1. INTRODUCTION

Wireless mesh network (WMN) is one of the wireless technologies to provide high-speed and cost effective wireless internet access service. WMN is self-configured, self-organized, and fault-tolerant to preserve the mesh connectivity after failure occurrence automatically. In fact, mesh nodes heal the failures by making instant and real-time decisions to find alternatives to transmit the data toward pre-defined destinations. All these features provide WMN with numerous overriding merits including flexibility, robustness, and reliability in service coverage, scalability to name a few in comparison with other similar technologies. These capabilities through relatively low initial cost, has attracted academia’s attention and became increasingly popular in business, industrial, and personal applications in the recent years. In general, three types of mesh nodes play role in formation of WMN’s architecture to provide the end user device with internet access: gateway, mesh router (MR) and mesh client (MC). Gateways serve the function of interconnecting the WMN MRs to the Internet. Wireless mesh backbone is formed by a number of stationary MRs. These intermediate nodes manage the network’s
traffics and are responsible for multi-hop connectivity provision among mesh hosts themselves and also between them and the Internet. Access points make such connectivity and as a result, make Internet access possible by forwarding MCs' traffics to MRs. The stationary or mobile MCs of different sub-networks such as Wireless Local Area Networks, worldwide interoperability for microwave access, or wireless sensor networks can connect themselves to the WMN's structure through the nearest MRs. Figure 1 represents the infrastructure of a given WMN.

Inter-channel interferences among neighboring nodes lead to dramatically capacity degradation and are of the most crucial challenges in traditional single-radio single-channel WMN. However, WMNs evolved to a new generation called single-radio multi-channels to exploit the advantage of using different non-interfering channels. However, the aforementioned problem had not been fully solved. Eventually, multiple radios have been proposed to switch on multiple channels to achieve higher throughput and to minimize channel conflict. By tuning the radio interfaces to the orthogonal channels, all interfaces can be utilized simultaneously with the minimum interference. Institute of Electrical and Electronics Engineers (IEEE) 802.11a standard operating on 5.8 GHz and IEEE 802.11b/g operating on 2.4 GHz presented 12 and 3 orthogonal frequency channels, respectively. Figure 2 depicts the available channels in IEEE 802.11 b/g.

Many communication mechanisms have been introduced in WMN including link scheduling [1,2], gateway placement [3,4], gateway selection [5,6], relay selection [7], broadcast routing [8–11], and multicast routing [12–16] to support its applications. A vital question in all these designing issues is how to assign the available channels to interfaces of nodes in such a way that the minimal interference is achieved. Figure 3 shows an example of conducted channel assignment (CA) in a given MRMC-WMN.

Because the network performance heavily depends on CA efficiency, considerable numbers of dynamic and intelligent CA schemes have been proposed so far. CA algorithms have been designed to address various criteria such as interference, routing, delay, connectivity, link scheduling, stability, and congestion to name a few [17–19]. These methods can leverage other algorithms like graph theory [20,21], game theory [22], and greedy approaches [23] to achieve the optimization goals. Generally, they can be categorized based on their main objective functions such as minimizing interference, maximizing throughput or congestion control [24–27]. Additionally, there is another classification including centralized [20,22–25,28–32] and decentralized [26,33–42] approaches. In centralized ones, benefiting from comprehensive knowledge of the whole network, a central controller makes CA's decisions. The central node as the assigner computes the most suitable channel according to received request and then, informs its neighbors as well as the assignee of newly calculated channel. Software-defined networking is a promising solution to design a wireless network in which a central controller makes decisions to guarantee a stable performance. There are studies to improve the approach, for instance, in [43] wireless network virtualization is considered as complement to software-defined networking’s management. On the other hands, there are no such central nodes in the decentralized methods; instead, each node can be both assigner and also assignee. It computes and selects its own channel independently and then announces its decision to others. As illustrated in Figure 4, in spite of diversity of CA schemes, they mostly include four main stages. In Figure 4, A, AE, and N depict the assigner, assignee, and neighboring nodes, respectively. CA algorithm is usually performed in two modes. Firstly, based on a pre-specified waiting time, the
algorithm is run periodically (usually long time intervals). Secondly, it can be triggered after occurrence of significant changes in the network topology, traffic or channel loads. For instance, a light-loaded channel at the beginning can be a loaded and congested one after a while leading to interference. There are studies to tackle the interference, for example, in [44], collision probability is modeled, and in [45], a hopping approach changes channel to another light-loaded to tackle the interference of pre-allocation channels on slotted carrier sense multiple access with collision avoidance (CSMA/CA).

(1) **Information gathering**: It is the initial phase of a CA algorithm. In this phase, nodes listen on channels and collect the required data from their x-hop away neighbor nodes in case of decentralized or all nodes in centralized fashion. The collected data can be related to bandwidth, channel usage, hop-count distance from the wired gateway, measured load, throughput of interfering neighbors, the assigned superimposed codes, and link status to name a few [46–52]. The comprehensiveness and accuracy of data driven by this phase has direct impact on the node’s ability to choose the best possible channel in the following decision-making step. Table I exemplifies the channel table of a given node in which, node 1 using channel 4 with measured load 100 is one-hop neighbor of a given node.

(2) **Channel assessment**: The given node computes the most proper channel based on the aggregated local data. It can consist of some sub-steps, for example, in [48], it is divided into three evaluation parts which will be explained in Section 3.1. Although, this computational phase differs in each proposed schemes, all have a common ultimate goal that channel $c$ should be selected so that it leads to higher network performance.

(3) **Channel notification**: The neighboring nodes are notified of the selected channel to update their tables.

(4) **Channel switching**: Interfaces are tuned to the selected channels. In [47], assigner is required to switch its channel only after receiving assignee’s approval as an acknowledgment message (ACK).

Regarding the importance of channel allocation, obviously selecting an improper channel is destructive and network throughput is affected seriously. It is crucial that all nodes play their defined roles in CA procedure so that their participations result in selection of best channel.

![Figure 3. Conducted channel assignment in an infrastructure of a Multi-Radio Multi-Channels-Wireless Mesh Network.](image1)

![Figure 4. The traditional channel assignment process.](image2)

<table>
<thead>
<tr>
<th>Node</th>
<th>Channel</th>
<th>Hops</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>300</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>2</td>
<td>120</td>
</tr>
</tbody>
</table>
However, in real networks, the trustworthiness of all participants including assigner, assignee and neighboring nodes are in doubt. Most of presented CA methods assumed implicitly that all nodes are well behaved, which is an ideal but impractical assumption. Indeed, their efforts could be easily wasted by a malicious mesh node. The presented study concentrates on the overlooked vulnerabilities in the majority of the CA algorithms causing some intrusions as well as the reported solutions against them in details.

The rest of this study is organized as follows. The existing attacks against CA algorithms are discussed in Section 2.1. Then, Section 3 surveys proposed anti-attack mechanisms. Finally, Section 4 concludes the paper.

2. ATTACKS AND VULNERABILITIES AGAINST CA ALGORITHMS

There are some series of attacks on the CA protocols that are originated from four main vulnerabilities: (i) Almost all CA schemes trust irrationally to the mesh nodes that they behave well intentioned and present information as it is. This wrong assumption makes them easy to exploit. The most efficient and intelligent CA models which do not consider malicious intentions and have no defensive mechanism against them, will expose sever network performance degradation. Although, the majority of proposed schemes ignore the negative impact of malicious mesh nodes, recently, a few works have addressed the possible security drawbacks. In [46,47], the possibilities of malicious assigner have discussed and in [48], some attacks were applied by the neighboring nodes. Additionally, the reliability of the assigner, the assignee, and the neighbor nodes has been challenged in [49–52]. (ii) There is no verification in the most CA procedure at all. Therefore, all kinds of nodes can violate the defined duties without any punishment or negative consequences. For instance, the fake information injected by the neighbors is used directly to accomplish CA computation improperly. Moreover, the lack of verification lets the malicious nodes choose an improper channel and notify others overtly. (iii) The probability of collusion among nodes have not been considered even in the suggested secure CA models. Most of them rely on a third trusted node verifying a received alarm message such as [46,47,49,51]. Thus, none of them can detect a security sabotage while the nodes collude to not send any alarm. (iv) The attacks taking their roots from the aforementioned problems could not be prevented even by the basic security frameworks such as authentication and encryption [46,51,53]. For instance, security protocols using encryption provide an adversary node with a secure communication channel to send their misleading information. It is not possible to assess the information cognitively.

These weak points increase the overall channel interference leading to frequent channel re-assignments and re-connections. In the following, different kinds of attacks caused by misbehavior of malicious mesh nodes will be explained and analyzed in details.

2.1. Attacks by neighbors

To our best knowledge, there are some attacks launched directly or indirectly by malicious neighbors that will be argued in this section.

2.1.1. Forged information attack.

Regarding its intention, the compromised node as neighboring node presents various fake, outdated, and unreal information. Figure 5 shows an example of the Forged information attack (FIA). The malicious node M lies about expected loads in channels 1 and 2 to newcomer nodes of G and D. Consequently, they choose these overloaded channels. This attack not only leads to improper channel decision, but it also deprives the victim node of regular data forwarding [48,51–53]. Some schemes are designed to assign the channels after sensing considerable changes in measured values such as signal-to-interference-plus-noise ratio, traffic load, and link availability to name a few. CA algorithm can be intrigued frequently to stop its normal procedure and re-calculate for a new channel repeatedly by sending wrong environmental information.

2.1.2. Altering the neighbor information attack.

An adversary may receive the information and forward its modification information to the neighbors and the channel assigners [51,52].

2.1.3. Increasing the neighbor set attack.

A malicious node increases the number of neighboring nodes by sending the base information on behalf of some phony nodes or even some theoretical nodes that are not neighbors at all owing to environmental factors [51,52].

2.1.4. Decreasing the neighbor set or packet delivery attack.

In this attack, an adversary as an intermediate node refuses to forward the data expected to be relayed toward an assigner with the aim of reducing the number of neighboring nodes [49,51,52].

2.1.5. Utilization-based conflict attack.

Normally, an assigner is expected to select a channel having the largest non-occupied portion called channel margin [50]. A malicious neighbor node intentionally manipulates the channels capacities to launch Utilization-based conflict attack (UCA). That is, it indirectly triggers the victim node to find another channel which causes less overall margin. Hence, further channel switching will be imposed on the victim. As shown in Figures 6 and 7, this attack can be applied in two scenarios: join UCA and leave UCA both through four a, b, c, and d steps. In the former, firstly, an adversary node M sniffs the aggregate channel margin of the target victim node V. Then, the node M uses...
the same channel (i.e., channel 2 in Figure 6b). Therefore, the node V observes a change in its local channel margin, and it runs its CA algorithm to find a channel with a higher capacity (i.e., channel 1 in Figure 6d). The UCA indirectly imposes a channel switching overhead on the victim node to keep it busy and suspend its data transmission.
In the latter attack, in order to influence the channel margin of the victim node V, the malicious node M switches its own channel to another (i.e., channel 3 from 1in Figure 7b). Therefore, the node M changes observed channel margin of the node V making it pause its data transmission (i.e., with node A) and get ready to run its CA algorithm. This attack would succeed if the victim node switches to another channel with less channel margin (i.e., channel 4 compared with channel 2 in Figure 7d). As it will be discussed later at Sections 2.1.6 and 2.2.5, this attack is part of other attacks named LIBA and Denial of data attack [50].

2.1.6. Link break attack.

Once a node changes its channel, it notifies its neighbors about the recent channel switching. The connection would be lost unless they update their channel table in a timely manner. This point is exploited by two attacks: direct Link break attack (LIBA), indirect LIBA. As it is depicted in Figure 8a, in the former attack, the malicious node M triggers node A to change its channel. Then, node A aims to inform node B to change the communication channel. This notification does not get to B because of occurred interference with M. Consequently, the victim nodes of A and B cannot agree on the new selected channel. As shown in Figure 9b, in the latter attack, a malicious node M makes two victim nodes B and C change their channels simultaneously (i.e., by UCA) with the object of breaking their link. In this scenario, because both end nodes change their own channels, they cannot hear the incoming notification messages [50].

2.1.7. Security alarm attack.

In some CA models like [47,52], neighboring nodes receiving abnormal and conflicting data send a message called security alarm message (SAM). Because of lack of verification, it can be exploited to damage a victim’s reputation and introduce it as a suspicious node. As an instant scenario, the node N gets ready to switch its own channel relieving the heavy traffic. The malicious node M broadcasts the fake SAMs pretending the misbehavior of node N. Because the SAMs are not verified, the gateway will assume the received requests or messages from N are invalid, and the channel cannot change.

2.1.8. Radio jamming attack.

This attack makes a channel disable by sending continuous or periodic network traffics or noise signals. It targets on disrupting the normal operation of the network and dropping its capacity [48,54].
2.2. Attacks by assigners and assignees

In addition to the discussed attacks by neighbors, there are some attacks launched by malicious assigners and assignees with the aim of network performance degradation that will be argued in this section.

2.2.1. Inappropriate channel assignment attack.

Under the normal CA operation, an assigner requires to assign the most efficient and the least loaded possible channel to its interfaces. However, a malevolent assigner deliberately selects an improper one. For instance, the channel can be chosen causing the highest interference [46,49,51,52] (e.g., allocating channel 1 by adversary node M in Figure 9a) or at least not causing the least possible interference [51]. Additionally, a malicious assigner assigns different channels to assignees intending to communicate with each other through a common channel. For example, channels 1 and 2 are assigned to nodes B and A in Figure 9b, respectively. Furthermore, the Inappropriate channel assignment attack (ICAA) can also be initiated once an assigner limits its choices to only a part of available non-overlapped channels [51,52].

Network Endo Parasite Attack (NEPA) [46] is another ICAA modeled in hyacinth algorithm [55]. The assigner not only chooses a high priority channel used by wired gateways, but it also does not notify other nodes other than its own children. It keeps the network unaware of the changes to increase the hidden usage of heavily loaded channel and finally aggravates interference in communication links over these channels. Figure 10 shows an example of the NEPA. Here, node M which is assumed to be a compromised node has assigned the channels on links MH and MI as the same channels used by links CG1 and DG1, respectively, which will raise the interference on links CG1C and DG1. Because the end points of C and G1 trust all nodes, they consider the high interference is the result of an external factor like noise.

2.2.2. Forging notification attack.

Notification phase can also be exploited in two ways [46,47,49,51,52]: First, a malicious node switches its interfaces and like the NEPA and the CEPA skips the notification step. Second, like Low-cost Ripple effect Attack [46], an adversary sends the forged and wrong channel number as a new assigned channel in order to tear down the links and increase the interference. The malicious node pretends to change its channel while it has not been changed in deed. In fact, it only sends misleading channel information as a new assigned channel. Because in this attack there is no need to actually change the channel, it is a low cost one. Figure 12 illustrates the Low-cost Ripple effect Attack. The compromised node M deceitfully informs its neighbors telling them that links MF and MH have been switched to new channels, although they are not changed. Based on these forged notifications, the neighbors change their assigned channel to links IJ, DE and, BD, which may change channels of A and C nodes too.

2.2.3. Improper switch attack.

An adversary assigner or assigne switches its channel to something different from notified or assigned channel [51,52].

2.2.4. Spam request attack.

In centralized CA approaches, an attacker assignee node requests a free channel, while actually it does not need it all [49]. Also, in decentralized CAs, each node as assigner can choose a channel and use it to transmit some unnecessary information with the aim of occupying it and preventing other nodes to access it.

![Figure 10. Interference enhancement on links CG1 and DG1 when node M assigns same channel 1 and 2 on links CG1 and DG1 to links’ children H and I in Network Endo Parasite Attack [56].](image-url)
2.2.5 Denial of data attack.

As shown in Figure 13, in this attack, a malicious node makes a victim node in normal-mode to go back to the information gathering or CA computation phase [50]. There are two ways for a malicious node to make this happen: DoD transition and catch-and-release DoD strategies. In the former strategy, the attacker intentionally tears down the connections; for example, an assignee or assigner leaves or changes its channel to disturb normal packet transmission. In the latter, an assignee or assigner causes the intentional failure through two steps; firstly, the LIBA is launched to break the victim’s connections to persuade it to use attacker’s channel. Afterwards, the malicious node does not transfer any data to be in silent mode. The objective of this approach is that a victim experiences link failure when it sends data packets to an attacker.

Finally, all the attacks discussed earlier may lead to ripple effect attack (REA). For example, this issue may occur once channel switching of a node leads to more switching across the whole network. Consequently, the network will slow down because all neighbors have to suspend their normal data packets transmissions responding to the recent changes. Therefore, the whole network would be tightly influenced by the REA in comparison with the RJA that only affects the attacker’s neighborhood [48]. The Figure 12 illustrates the effect of the attack by the arrows.

As it is being discussed, there are some attacks on CA algorithm in MRMC-WMNs. Figure 14 shows a
chart briefing of the presented attacks. However, a verifier node that has been proposed recently as a defensive mechanism (explained in Section 3.1) is assumed reliable and attacks related to malicious verifier are overlooked so far. Moreover, there are some more complex attacks in which mesh nodes unify against a victim node, such as collusion of the assigner and the assignee or the neighboring nodes and the verifier to name a few. Table II compares security based studies on CA algorithm in which different nodes are considered to be malicious.

3. ANTI-ATTACK STRATEGIES VERSUS CHANNEL ASSIGNMENT MECHANISMS

As it was explained in Section 2, CA algorithms suffer from lack of verification and mesh nodes can be compromised based on their roles. Thus, most of the reported solutions have augmented a kind of verification phase to monitor the behavior of mesh nodes with the aim of controlling information accuracy and independent decisions. These verification mechanisms will be explained in the two following sections.

3.1. Solutions against malicious neighbors

3.1.1. Forged information attack defensive mechanism.

The proposed CA algorithm in [32] applies two steps against FIA. Initially, two-hop mesh nodes exchange the information of their channels or expected load on their links. Afterwards, the receiver node compares the gathered neighbors’ information. Inconsistent information triggers the receiver node to ask a trusted node about the suspected neighbors. The trusted node is selected based on the highest average degree of trust value calculated from packet drop, packet loss, and CA attacks criteria. The reference [52] also presents another comparison based protocol. Firstly, the received messages are verified by one-hop neighbors. Second, when no anomaly is found in the verification phase, a positive control message would be sent to notify the message is credible. Otherwise, a SAM is sent to notify its abnormality and a channel scan is requested. A low-load node takes charge of scanning which is randomly selected by a randomized algorithm called scanner selection strategy. Once the message is again confirmed to be illegal, a SAM is sent to notify that it should be discarded. The reference [48] proposed protocol in which a three-step process (channel evaluation, channel switch cost evaluation, and randomized selection)
neighboring node checks if packet time of short messages during dedicated time slot. The receiver is required to send its immediate neighboring nodes several property. Before information gathering phase, every node mechanism.

3.1.3. Increasing the neighbor set attack defensive mechanism.

In the references [32,52] two-hop or one-hop neighbors’ assigner extract the inconsistency and then a trusted verifier check it again. They are different in the verifier’s selection strategy. However, in [48], it is the assigner endeavoring to tackle FIA in a different perspective which is not based of abnormality. Instead, considering less weight multiple, the assigner always makes the two-hop nodes information less influential.

3.1.2. Altering the neighbor information attack defensive mechanism.

The references [51,52] prevent the information from alteration by cryptography approaches or digital signature.

3.1.3. Increasing the neighbor set attack defensive mechanism.

In [51,52], it is proposed that the assigner verifies the neighboring node’s message based on its physical property. Before information gathering phase, every node is required to send its immediate neighboring nodes several short messages during dedicated time slot. The receiver checks if packet time of first short message meets expected transmission time regarding the distance between them. Also, any duplicated message is checked to detect relay attack. The receiver node maintains a table of the neighbors including their IDs, distances, and average of received messages’ physical properties such as signal strength indicator (RSSI). Nodes that do not send any short message during their pre-assigned time slot will be removed from the table. It is claimed that in this step, a real neighbor set has been formed on the basis of accurate physical measurement. An example of this set is illustrated in Table III. Afterwards, during information gathering phase, the assigner decrypts the encrypted or the signed message of the sender and checks the existence of sender’s ID in its neighbor set. Then, regarding the pre-specified threshold it checks the acceptable difference between the calculated averaged RSSI and the recent RSSI to distinguish between theoretical and realistic neighbors and tackle the Increasing the neighbor set attack.

3.1.4. Decreasing the neighbor set attack defensive mechanism.

The reference [32] proposes a solution against the DNSA in which a receiver node sends ACKs for the sender through various paths. When all the ACKs arrive, if any of them looks different, there has been a DNSA. When the sender detects the attack, the packet forwarding would be stopped through the malicious node. Also, the sender updates its neighbor behavior table and drops the malicious node’s trust level. As an example, Figure 15 shows three ACKs sent through different paths from the sender C to the receiver A, where the malicious node M’s ACK1 would disclose its attack.

<table>
<thead>
<tr>
<th>CA algorithm</th>
<th>Malicious neighbor</th>
<th>Malicious assigner</th>
<th>Malicious assignee</th>
<th>Malicious verifier</th>
<th>Nodes collusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>[46]</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[47]</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[48]</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>—</td>
<td>No</td>
</tr>
<tr>
<td>[49]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[50]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>—</td>
<td>No</td>
</tr>
<tr>
<td>[51]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>[52]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

CA, channel assignment.

Table II. The malicious nodes in the studies on CA’s security.

<table>
<thead>
<tr>
<th>ID</th>
<th>Reception condition</th>
<th>Distance (m)</th>
<th>Average of RSSI (dBm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>10</td>
<td>-30</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>50</td>
<td>-45</td>
</tr>
<tr>
<td>3</td>
<td>—</td>
<td>50</td>
<td>—</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>30</td>
<td>-45</td>
</tr>
</tbody>
</table>

RSSI, received signal strength indicator.

Table III. An example of the neighbor table of a given node [51].
spread spectrum. To be protected by jamming-resilient techniques like a greedy algorithm. Also, the routes should be reorganized both locally and globally to bypass the RJA occurs, the channels should be changed based on a pre-decentralization. In fact, it is assumed the channel is receive any base information is put into prohibited (Figure 9), when node A compare all received Acknowledgments. Ack, acknowledgment message.

3.1.5. Utilization-based conflict and link break attacks defensive mechanism.

In [50], defense mechanism against the UCA and the LIBA includes three steps. In the first step, the neighbor nodes that cause a channel switching which in turn escalates connectivity and throughput issues are supposed to be suspicious. The information gathered from these nodes will be less influential. Second, an assigner selects a set of candidate channels. Current assigned channel will not be changed if it is one of the candidate channels, even in the present of a better one. Third, a channel becomes eligible for changing only if multiple neighbor nodes suggest so.

3.1.6. Security alarm attack defensive mechanism.

Random selected verifier nodes verify validity of the positive or negative control messages [52]. According to its misbehavior node filtering strategy liar nodes had presented false-positive or false-negative in the previous round would be filtered to detect the Security alarm attack.

3.1.7. Radio jamming attack defensive mechanism.

The protocol presented in [48] claims that it is resistant to the RJA in two ways. First, an attacker has to jam all channels used to gather base information to make the protocol disable and launch the RJA; therefore, it would not be a low-cost attack. Second, this protocol divides channels into two lists (the preferred and the prohibited) based on evaluation in the first step. The suspicious channel on which the node cannot receive any base information is put into prohibited channel category. In fact, it is assumed the channel is under the RJA and hence, it should be prohibited during a pre-defined time. Another joint defense mechanism against the RJA has been presented in [54]. When the RJA occurs, the channels should be changed based on a greedy algorithm. Also, the routes should be reorganized both locally and globally to bypass the attack. Finally, in [51], the physical layer is suggested to be protected by jamming-resilient techniques like spread spectrum.

3.2. Solutions against malicious assigner and assignee

3.2.1. Inappropriate channel assignment attack defensive mechanism.

The authors presented NEPA and CEPA in [46], and they believe that CA coupled with channel scanning and a distributed Intrusion Detection System can reveal these ICAAs. After dividing the network into zones, randomly selected controllers within each zone are responsible for verifying announcements. This study assumes that because channels are assigned to the radios for fairly long periods and switched upon a significant change in network, the overhead scanning is negligible. Afterwards, the authors present another detective and preventive solution against the aforementioned attacks in [47]. In this idea, smart children add a security mechanism and play the role of verifiers in hyacinth model. Special tree-based hyacinth structure has made it highly practical [47]. Each node equipped with two separate network interfaces (NICs) can be a parent and also a child of its parent. The node communicates with its parent by its UP_NIC interface and controls its children over DOWN-NIC interface. Although, in the normal hyacinth procedure, every child as assignee should obey its parent decision and has no intelligence, in this study, the parent depends on its smart child to assign the channels because it needs child’s approval message to assign a channel. The child node calculates the approximate channel based on the information provided by its parent node and compares it with the requested channel tuning in the notification message. In case of equality, it replies with a positive ACK message permitting the parent node to change its channel. Otherwise, a negative ACK message would be sent to prevent the channel from changing (no independent decision). The authors in [49] came up with a cluster-based CA method handling the ICAAs. First, it considers two criteria to detect the ICAA attack in which a common overloaded channel is allocated to all assignees (Figure 9a); the first criterion is related to the number of interfaces and assigned channels per node that should be exchanged among the nodes. Also, the second one is associated with trust degrees. Each node updates a personal history table occasionally including the packet loss, packet drop, and neighbors’ behavior columns to compute the trust. Using these criteria, the adversary node is believed to be able to distinguish from the node that actually is allowed to use the only available channel. Second, the study alleges that the ICAA attack shown in the Figure 9b is avoidable, if assignees A and B reach an agreement over allocated channel before communicating with each other. They exchange and compare their channels received from the common assigner M. Obviously, the assigner would know as adversary in case of detecting any different channels. In [51], a trusted entity such as a gateway node selects randomly a verifier node which is responsible for monitoring the assigner’s behaviors and decisions to detect the ICAA. There is a direct relationship between the number...
of verifiers and the security level. Also a tradeoff between them and resultant overhead should be considered. Both the assigner and the verifier compute the channel separately based on the same CA algorithm and information. Then, the assignee receives the assigner’s notification and also verifier’s message including computed channel as confirmation message. The receiver checks two expectations, first, the verifier authorization in its zone and second, the equality of channel in the notification and confirmation message. In the end, the node accepts newly assigned channel. In [52], the neighbors of an assigner verify the tuned channel in the first, second, and the third rounds of verifications as shown in Figures 16, 17, and 18, respectively. In these Figures, channel usage message refers to the base information exchanged among nodes. Also, channel change message refers to notification message. In addition, NBR is the node theoretical neighbor set.

After receiving a notification including the list of neighbors of the assigner, the node will conduct the first-round verification as shown in the Figure 16. It will then collect base information of the assigner’s neighbors, computes the channel, and compares the results with the incoming notification. If the node finds no difference, the notification will be effective. Otherwise, a SAM is broadcasted, and the second-round verification will be triggered as shown in the Figure 17. The second-round verification procedure is similar to the first round except here; a random node computes the channel based on the proposed verifier selection strategy. The idea behind the verifier selection strategy algorithm is closer nodes to assigner have higher chance to be chosen. If the second-round cannot find any abnormality, the channel would be assigned. Otherwise, the notification message would be discarded. Nodes that did not receive second-round verification result during predetermined time run a third-round as shown in the Figure 18.

As it is mentioned, in references [46, 51], a random verifier, in [47, 49], the assignee (smart child in [47]), and in [52] a random verifier and the assignee are in charge of channel computation and verification. In comparison, the presented algorithm in [52] considers more verification layers and also the filtering strategy as an overriding advantage (Section 3.1.6).

3.2.2. Forging notification attack defensive mechanism.

The reference [46] believes that monitoring approaches like channel scanning and Intrusion Detection System mentioned in Section 3.2.1 can also reveal the Forging notification attack (FNA) which forged notifications are sent in. In addition, the reference [47] changes the normal hyacinth procedure in which the nodes have to accept the

Figure 16. First round verification [52]. CCM, channel change message; CUM, channel usage message.
information as they receive. In this study, nodes modeled in the hyacinth procedure are smart and children individually are able to verify the notification to detect any abnormality. For example, as shown in Figure 19, the node C1 will notice the discrepancy in the received message from the parent node M when it compares its current channel $i$ tuned on radio 2 and received assigning channel $k$ to interface 2. After finding such discrepancy, it sends a monitor request message containing the suspicious parent’s ID and the attributed misinformation. Nodes maintain a bad-credit value representing the degree of misbehaving which is measured based on received monitor request messages ($0 \leq \text{bad credit} \leq k$). Value of 0 and $k$ mean the node is well behaving and misbehaving, respectively. The received messages with bad-credit of higher than 0 would be discarded. This reference claims that a child or a neighbor cannot maliciously accuse a parent, because nodes are monitored by their own smart children. Furthermore, the study alleges that with the help of this detective mechanism, malicious messages won’t initiate any unnecessary CA algorithm and consequently, no REA threatens the network. Additionally, the smart children are claimed to be not only able to detect the attack but they can also prevent them.

Finally, the FNA is believed detectable by monitoring packet loss factor in [49]. Because, there would be an extreme increase in the factor rate maintained in the history table and therefore, the attacks will be revealed.

However, all suggested approaches against FNA are detective not preventive and the approach in [47] is limited to the hyacinth model.
3.2.3. Spam request attack defensive mechanism.

The cluster-based algorithm in [49] assumes that the spam requests of malicious assignees having no private and session keys of the authenticated cluster members will be rejected by the cluster heads.

3.2.4. Improper switch attack defensive mechanism.

In [51], the Improper switch attack claimed to be tackled by the randomly selected scanner chosen in the same manner as the suggested verifier to check the appropriation of conducted channel switching. In [52], the node that cannot communicate over newly assigned channel asks the random other side’s neighboring nodes to scan the assigned channel to verify the switching phase. In comparison, the approach in [52] seems more efficient because the verification is done only in necessary cases.

Table IV summarizes attacks on CA algorithms and presented solutions.

4. DISCUSSION

There is a broad range of security attacks designed to fulfill different goals such as disclosure of confidential information to name a few. The main objective of all mentioned CA attacks in this study is dropping network performance in various ways. The victim nodes directly or indirectly are imposed unnecessary CA computations and channel re-assignment. Although, the solutions attempt to detect attacks and minimize their influence on the network, they themselves have computational load. There is a tradeoff between intended security level and the complexity of the defense mechanism. For instance, the references [51,52] with higher complexity cover more attacks and a random verifier monitors other nodes decisions. However, as it is simulated in papers, despite the computational expenses, the model designed in [52] show superlative goodput in comparison with the previous works. Therefore, in meeting the aforementioned challenges, an optimal smart CA should be developed to provide more security level by minimum computation.

Although, the aforementioned approaches provide extra protection, they have some flaws. Firstly, none of illustrated works has a comprehensive secure protocol against all presented attacks. Secondly, there are some possible collusion attacks that all presented models do not consider them at all. In a likely scenario where malicious neighboring nodes collude and send the same forged messages, the aforementioned compared based mechanisms [32,52] are unable to detect any inconsistency and to avert the FIA. Additionally, the presented smart child [47] and its parent can collude not to send monitor request messages enhancing the chance of broadcasting forged information or to select an overloaded channel to increase interference throughout the network (e.g., nodes F and I in Figure 20). In addition, the neighbors or assigners can collude with the verifier or the scanner node to accept the same forged information or inappropriate channel to name a few. Therefore, presented verifier based ideas in [32,46,47,49,51,52] to address FIA, Security alarm attack, ICAA, or Improper switch attack should consider the collusion attack too. Although, the risk still exists, the probability of collusion with malicious verifier would be decreased if the minimum number be considered two. Once two or more verifiers or scanners present different opinions, a mechanism should be defined to choose the applicable one among different opinions.

Thirdly, the proposed verifier nodes that are responsible for verifying suspected nodes are not itself chosen properly and there will be no way of verifying its decision. In the references [46,49,51,52], they are chosen randomly among ordinary nodes of network. Therefore, a malicious node has an equal chance of being selected as well. In [32] they are selected based on some criteria such as packet drop and other CA attacks, not the honesty of their recommenders. It is possible for a malicious node to behave genuinely in order to gain the highest trust degree. The malicious node then gives forged opinions when they are selected and asked about victim neighbors. Also, the smart child node in [47] as reliable...
<table>
<thead>
<tr>
<th>CA Coordination</th>
<th>Network structure</th>
<th>Algorithm to model attacks and solutions</th>
<th>Trusted verifier</th>
<th>Presented attacks</th>
<th>Proposed solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>[46] Distributed MRMC</td>
<td>Hyacinth</td>
<td>Random node</td>
<td>NEPA</td>
<td>IDS</td>
<td>• Channel Scanning</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CEPA</td>
<td></td>
<td>• Smart child mechanism</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LORA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[47] Distributed MRMC</td>
<td>Hyacinth</td>
<td>Child node</td>
<td>NEPA</td>
<td></td>
<td>• Channel evaluation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CEPA</td>
<td></td>
<td>• Random channel selection</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LORA</td>
<td></td>
<td>• Channel switch cost evaluation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Assigning higher weight to one-hop vs two-hop nodes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Classification channels into preferred and prohibited channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[48] Distributed MRMC</td>
<td>Systematic based CA algorithm</td>
<td>No trusted node</td>
<td>REA</td>
<td></td>
<td>• Recording packet loss factor in a history table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RJA</td>
<td></td>
<td>• Trusted node verification (it is not chosen randomly)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FIA</td>
<td></td>
<td>• Exchanging the number of interfaces and assigned channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Arriving assignees at an agreement over common assigned channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[49] Centralized MRMC</td>
<td>CCA</td>
<td>Trusted and cluster head nodes</td>
<td>FIA</td>
<td></td>
<td>• Comparing ACKs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DNSA</td>
<td></td>
<td>• Monitoring channel request by cluster head</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ICAA</td>
<td></td>
<td>• Recording packet loss factor in a history table</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FNA</td>
<td></td>
<td>• Trusted node verification (it is not chosen randomly)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SRA</td>
<td></td>
<td>• Exchanging the number of interfaces and assigned channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Arriving assignees at an agreement over common assigned channel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[50] Distributed MRMC</td>
<td>Common operational based CA algorithm</td>
<td>No trusted node</td>
<td>UCA</td>
<td></td>
<td>• Detecting suspicious node to make them less influential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LIBA</td>
<td></td>
<td>• Selecting new channel among a set of candidate channels</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>DoDA</td>
<td></td>
<td>• Changing channel only if it is triggered by multi nodes rather than one</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[51] Distributed MRMC</td>
<td>Hyacinth</td>
<td>Random node</td>
<td>ISA</td>
<td></td>
<td>• Digital signature and MAC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INSA</td>
<td></td>
<td>• Measuring the physical properties of messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ICAA</td>
<td></td>
<td>• Verifying CA, notifications and channel switching by randomly selected verifier or scanner</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ANIA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>RJA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[52] Distributed MRMC</td>
<td>Hyacinth</td>
<td>Random node</td>
<td>SAA</td>
<td></td>
<td>• Digital signature</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ISA</td>
<td></td>
<td>• Filtering strategy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>FIA</td>
<td></td>
<td>• Measuring the physical properties of messages</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>INSA</td>
<td></td>
<td>• Verifying CA, notifications and channel switching by randomly selected verifier or scanner after receiving the SAM</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ICAA</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>ANIA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ACK, acknowledgment message; ANIA, Altering the neighbor information attack; CA, channel assignment; CCA, Cluster-base CA algorithm; CEPA, Channel Ecto Parasite Attack; DNSA, Decreasing the Neighbor Set Attack; DoDA, Denial of data attack; FIA, Forged Information Attack; FNA, Forging Notification Attack; ICAA, Inappropriate Channel Assignment Attack; IDS, Intrusion Detection System; INSA, Increasing the neighbor set attack; ISA, Improper switch attack; LIBA, Link Break Attack; LORA, Low-cost Ripple effect Attack; MRMC, Multi-Radio Multi-Channels; NEPA, Network Endo Parasite Attack; REA, Ripple Effect Attack; RJA, Radio Jamming Attack; SAA, Security alarm attack; SAM, security alarm message; SRA, Spam request attack; UCA, Utilization-based Conflict Attack.
verifier is not as trusted as it is assumed. A malicious child as an assignee that should verify the parent’s CA may refuse a suitable requested channel in order to deprive its parent of a better channel. Additionally, as it is explained, a verifier node can be part of a collusion attacks.

Finally, some studies apply unrealistic assumptions. For example, in [48], it is assumed that one-hop nodes are more reliable than the two-hop ones and a higher weight is considered for their opinions. However, there is no reason to assume that two-hop nodes are more prone to present fake information. In reality an adversary can be an immediate neighbor that in this case this model gives a higher weight to attacker. Also, the authors in [46] assume that channel scanning is done rarely and so its overhead is negligible. However, this ideal assumption is unrealistic and impractical in large scale [51,52]. As it is explained in the Section 2.1, malicious nodes send forged information to intentionally intrigue victim nodes to perform CA frequently. The proposed approach in [49] assumes that real members of a cluster do not send a spam request. In reality, these so-called actual members of cluster having the session keys can also send spam request too, and the proposed approach cannot detect the members’ Spam request attack. Finally, the protocol in [51] measures physical values of neighbors’ messages. However, the approach does not seem to verify the context and correctness of messages to handle the FIA. In fact, it supposes that actual neighbors are not malicious nodes. In fact, malicious nodes can be the neighbor nodes satisfying all expected physical attributes or even signing its forged messages. Therefore, the physical parameters of messages cannot be the precise criterion of verification as such. However, it would be a solution for Increasing the neighbor set attack.

According to the mentioned uncovered vulnerabilities, more complex and multistage attacks can be presented. Because most of these threats take their root from lack of real cognition of neighbor, assigner, assignee, and verifier nodes, a trust-based CA can improve current defensive approaches. Trust and reputation models are being used to enhance the accuracy of decision-making process. It can be added to the CA traditional algorithm in order to make the CA smarter to select the most trusted verifier and consequently, the most efficient channel.

Reference [49] designs a simple history table to compute trust values. However, the utilized history table can be constructed more precisely with considering various kinds of attacks measured by mathematical analyses and functions like time functions instead of blindly using of positive or negative marks for limited items. In this study, once a node randomly observes a good behavior of another node, it adds a positive mark in its table related to the monitored node. This general scoring strategy makes the system unsafe because for instance, a node may behave well in preventing packet drop to increase its trust level enough to launch others CA attacks in a reasonable time. In addition, a newly connected node has an empty history table. Because the trusted node is chosen by analyzing this table, the new node has no way of choosing the trusted node until enough time passes and the table fills with some entries. An exact and comprehensive trust and reputation model promotes security of the CA protocol.

Finally, although all mentioned current CAs’ are targeting to reduce interference, there is a significant demand of detailed analytical compare studies on influence of presented security attacks and CA solutions on network performance in real-time applications. Kinds of performance metric such as delay, throughput, capacity of connection and bandwidth utilization can be modeled [57] to evaluate and compare the adverse impact of attacks and solutions’ efficiency and complexity.

5. CONCLUSION

The main research issue in multi-radio multi-channel wireless mesh networks is proper assignment of channels to the radio interfaces, which can be conducted by using CA
strategies in order to reduce the overall network interference. In spite of existing of a rich literature of CA algorithms, most of them pay no mind to security menaces and its damaging effects. In fact, in spite of the precise design of proposed CA algorithms, the malicious nodes have opportunity to negatively influence network performance in many ways. However, recently, many CA attacks and some defensive mechanisms have been considered which were surveyed in this study. In general, the attacks can be categorized into two categories: attacks implied by a neighbor or an assignee of malice, which all take their roots in absence of any veriﬁcation in CA algorithms due to unreasonably trusting on mesh nodes. Even the reported secured CA policies do not provide an exhaustive solution. Current solutions’ open issues are overlooking different kinds of collusion attacks, considering some nonrealistic assumptions to simpliﬁcation, inaccurate selection and monitoring of veriﬁer nodes and ﬁnally lack of a complete scheme meeting all requirements of a secure CA in real case. In conclusion, CA algorithms in WMNs need to be more evolved with a focus on security.

6. NOTATIONS

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMN</td>
<td>Wireless Mesh Network</td>
</tr>
<tr>
<td>LIBA</td>
<td>Link Break Attack</td>
</tr>
<tr>
<td>MRMC</td>
<td>Multi-Radio Multi-Channels</td>
</tr>
<tr>
<td>RJA</td>
<td>Radio Jamming Attack</td>
</tr>
<tr>
<td>CA</td>
<td>Channel Assignment</td>
</tr>
<tr>
<td>FNA</td>
<td>Forging Notification Attack</td>
</tr>
<tr>
<td>NEPA</td>
<td>Network Endo Parasite Attack</td>
</tr>
<tr>
<td>CEPA</td>
<td>Channel Ecto Parasite Attack</td>
</tr>
<tr>
<td>FIA</td>
<td>Forged Information Attack</td>
</tr>
<tr>
<td>REA</td>
<td>Ripple Effect Attack</td>
</tr>
<tr>
<td>DNSA</td>
<td>Decreasing the Neighbor Set Attack</td>
</tr>
<tr>
<td>ICAA</td>
<td>Inappropriate Channel Assignment Attack</td>
</tr>
<tr>
<td>UCA</td>
<td>Utilization-based Conflict Attack</td>
</tr>
</tbody>
</table>

REFERENCES

16. Jahanshahi M, Dehghan M, Meybodi MR. A mathematical formulation for joint channel assignment and multicast routing in multi-channel multi wireless mesh


