Multimedia Tools and Applications (2019) 78:29899–29920
https://doi.org/10.1007/s11042-018-6684-z

A novel countermeasure technique for reactive jamming attack in internet of things

Alaba Ayotunde Fadele[1] · Mazliza Othman[1] · Ibrahim Abaker Targio Hashem[2] · Ibrar Yaqoob[3] · Muhammad Imran[4] · Muhammad Shoaib[4]

Received: 24 May 2018 / Revised: 30 August 2018 / Accepted: 12 September 2018 /
Published online: 25 September 2018

Abstract

In recent years, Internet of Things (IoT) has attracted significant attention because of its wide range of applications in various domains. However, security is a growing concern as users of small devices in an IoT network are unable to defend themselves against reactive jamming attacks. These attacks negatively affect the performance of devices and hinder IoT operations. To address such an issue, this paper presents a novel countermeasure detection and consistency algorithm (CDCA), which aims to fight reactive jamming attacks on IoT networks. The proposed CDCA uses a change in threshold value to detect and treat an attack. The algorithm employs channel signal strength to check packet consistency by determining if the data transmission value contradicts the threshold value. The node that sends the threshold value is periodically checked and the threshold value is compared with the current value after data transmission to find out if an attack has occurred in the network. Based on realistic simulation scenarios (e.g., with varying traffic interval, number of malicious nodes, and random mobility patterns), the performance of the proposed CDCA is evaluated using a Cooja simulator. Simulation results demonstrate the superiority of the proposed technique compared with contemporary schemes in terms of performance metrics such as energy consumption, traffic delay, and network throughput.

Keywords Internet of things · Network security · Jamming attack · Countermeasures

1 Introduction

Internet of Things (IoT) is used in various application domains, such as smart cities, intelligent transportation systems, and healthcare [32]. Protecting the users of these applications is a

✉ Alaba Ayotunde Fadele
ayotundefadele@siswa.um.edu.my

✉ Mazliza Othman
mazliza@um.edu.my

✉ Ibrahim Abaker Targio Hashem
ibrahimabaker.targiohashem@taylors.edu.my

Extended author information available on the last page of the article

demanding task because each application has different security requirements [19, 29, 35, 37] and harmful attacks may affect network performance. The vulnerability of IoT networks to various types of jamming attacks draws attention to the importance of security [3, 22]. Securing these networks is crucial because of the characteristics and features of IoT devices [20, 21]. Conventional cryptographic security mechanisms have been introduced into the wireless sensor domain to protect IoT applications from jamming attacks such as denial of service (DoS), network spoofing, and packet injection [27]. However, these mechanisms may be impractical for IoT, mainly because they are unsuitable for resource-constrained devices. Thus, IoT remains susceptible to jamming attacks that target various communication media [9, 38].

Jamming is an important class of DoS attack [14], that may employ malicious nodes to disrupt legitimate communication between IoT devices through intentional interference. These attacks negatively affect the performance of the resource-constrained small devices, thereby harming the network [22]. In these attacks, the objective of an adversary is usually to drain the limited energy resources of small devices by halting communication [10]. Jamming is one of the most harmful attacks that can obstruct wireless communication channels by presenting counterfeit packets and disrupting the radio communication frequencies in networks. Therefore, such an attack is a major threat to IoT networks that consist of nodes with limited energy and resources [32]. Jamming is performed either consistently or intermittently [2]. Reactive jamming is an important type of attack on a physical and medium access control (MAC) layer; such an attack causes the energy of the small devices to be consumed unnecessarily during communication. The performance of resource-constrained networks depends mainly on how efficiently their resources are used in [8, 20].

The modeling of the security attacks helps to understand an actual view of jamming attack in IoT networks and enable us to decide the mitigation plans [13, 37]. The reactive jamming attacks security modeling is illustrated in Fig. 1, which shows the reactive jamming attack security modeling and the network vulnerability to attacks. All devices in IoT have low memory and limited computation resources, making them vulnerable to resource enervation attack. Attackers can send messages or requests to a specific device to consume their resources. DoS (Jamming) attack is also possible due to man-in-the-middle attack [5, 25].

Reactive jamming is the most difficult and challenging type of attack to detect in an IoT network. It is more dangerous than constant, deceptive, and random jamming attacks in terms of performance [9]. In reactive jamming, an attacker turns a legitimate node into a jammer by transmitting signals only when it detects that a normal node is sending packets to other devices; otherwise, the node remains idle [23]. Therefore, introducing a robust and compatible approach to tackle reactive jamming in IoT networks is imperative.

The contributions of this study are as follows: We investigate various countermeasures for reactive jamming attacks and propose an efficient defense mechanism called countermeasure detection and consistency algorithm (CDCA). The proposed algorithm detects an attack based on channel signal strength, packet inconsistency, and node location. A reactive jamming model is formulated and simulation results demonstrate the superior performance of CDCA compared with contemporary schemes in terms of metrics such as energy consumption, traffic delay, and network throughput.

This paper is organized as follows. Section 2 reviews related studies. Section 3 discusses the system model. The proposed CDCA mechanism is described in Section 4. Section 5 presents the implementation details. The simulation results and analysis of CDCA are discussed in Section 6. We provide the conclusion in Section 7.

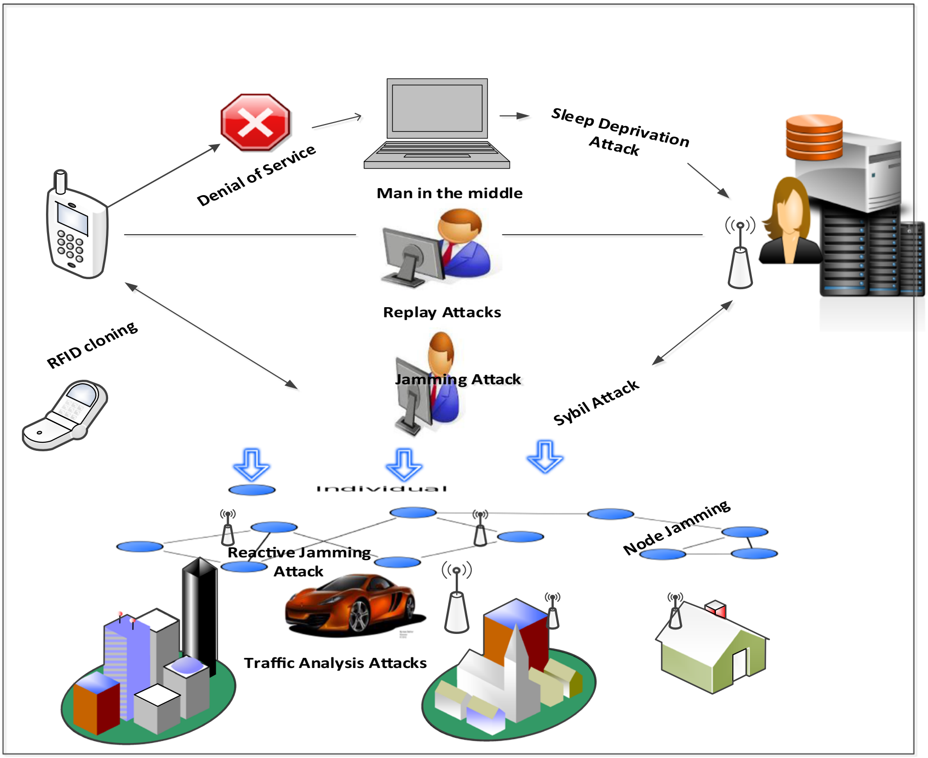


Fig. 1 Reactive jamming attack security modeling in IoT

2 Related works

Many security countermeasures employing a variety of defense techniques have been developed to address jamming attacks, including reactive jamming, in the IoT domain. Securing IoT networks from jamming attacks is not a new problem and has been extensively studied. As mentioned, jamming is a DoS attack [11, 14] that employs malicious nodes to disrupt legitimate communication between IoT devices through intentional interference. Jamming attacks have been investigated in various domains. For example, the study [11] proposed a mechanism to counter such an attack in cluster-based wireless sensor networks (WSNs); the researchers investigated the behavior of a jamming attack and analyzed its effect on the performance of WSN using cyclic redundancy check (CRC). The analysis shows that the performance increases by approximately 30% more than that of normal traditional reactive jamming attack approaches such as packet delivery ratio (PDR) developed in [28] and retreat from interference in [34]. The traditional approaches depend on CRC of packets to determine if correct packets are received. However, these approaches cannot differentiate between packet failures due to weak radio links and interference. Furthermore, when reactive jamming occurs, evaluating an accurate/exact PDR is infeasible through the traditional approach because packets are sent only when the jammer detects activities in the channel.

A novel anti-jamming strategy for IoT systems was proposed in [26] using an IoT controller to protect devices against a malicious radio jammer. The Colonel Blotto game (Nash equilibrium) with continuous and asymmetric resources was also employed to model the jamming attack. The drawback of this approach is the high energy consumption and traffic delay caused by the malicious nodes in the network.

Energy-efficient routing for the jamming attack presented in [33] adopted frequency hopping and frame masking. To improve energy conservation at the nodes, the proposed mechanism divides packets into fragments. The proposed mechanism requires dedicated hardware, which increases its implementation cost besides energy consumption and delay issues.

A solution for jamming suppression in IEEE 802.11b networks was provided in [30] using an electronically steerable parasitic antenna radiator. The results show that connectivity during a jamming attack can be improved by using switched-beam antenna enhancing system bandwidth. The major challenge is network degradation as the number of malicious nodes increases, leading to an increase in energy consumption and delay.

A jamming attack detection and countermeasure based on the ant system was proposed in [12]. This approach used the ant algorithm based on mobility to prevent and detect jamming attacks on IoT. However, this approach tends to increase delays because it requires multiple access points, which in turn increase the network complexity and implementation cost.

To detect jamming attacks, a threshold-based jamming countermeasure (TJC) scheme was proposed in [20]. The proposed technique primarily relies on a packet count threshold value, which is determined and sent by the base station (BS) to all nodes in advance. An attack is detected by the BS if a node sends more data than the threshold value. Unlike WSN where nodes are often homogeneous and where consistent data are sent, IoT involves heterogeneous Internet-connected devices with variable data transmissions in [7]. Therefore, TJC may be unsuitable for IoT. Although IoT devices are assigned with IP addresses and their location can easily be tracked. Unlike TJC, our proposed CDCA algorithm exploits IP in [27] and location-based authentication besides data consistency to counter a jamming attack, which is not the case with WSN.

Based on the drawbacks observed from the existing techniques, CDCA is proposed as a countermeasure for reactive jamming attacks. This technique checks packet detection strength and node location consistency when reactive jamming occurs in an IoT network. The packet detection strength is monitored to find out if any change has occurred between the packet signal strength and the threshold value. The node location consistency keeps the location information of legitimate neighboring nodes with their fixed threshold value, which is constantly checked at time intervals. Unlike contemporary schemes that are centralized, CDCA is a distributed approach that can operate on all nodes.

Reactive jamming is a type of jamming that turns a normal node into a jammer by transmitting signals only when an attacker detects that the normal node is sending packets to other nodes, else it keeps quiet [5, 12]. This makes reactive jamming difficult to detect unlike the other types of jamming attack such as constant, deceptive, and random jamming attacks. A constant jammer emits radio signals continuously in the communication channel. The goal of a constant jammer is to continuously emit signals to keep the communication medium busy, thereby preventing legitimate nodes from accessing the communication channel. A deceptive jammer continuously sends and inserts random data into the communication channel rather than sending random bits without gaps between the packet broadcasts. In deceptive jamming, no effective communication occurs between nodes as a result of constant data streaming performed by a deceptive jammer in the channel. A random jammer behaves in a manner similar to constant and deceptive jammers. The unpredictable behavior of a random jammer causes difficulty in detecting its attack compared with constant or deceptive jammers [24]. Random jamming is more intelligent than constant and deceptive jamming because it conserves energy given unlimited power supply. The jammer node alternates between sleeping and jamming after a regular interval.

Recently, IoT has gained popularity because of the increase in applications that are connected to the Internet. Researchers [18, 24, 30] have predicted that the number of connected devices by 2020 will be four times more than the human population; for users to develop trust in IoT networks, jamming attacks, especially reactive jamming, have to be addressed. As discussed in Section 1, reactive jamming remains the most difficult attack to detect in IoT networks. Moreover, in the innovation of WSNs, jamming attacks are the main security issues that have to be addressed before users can gain trust in these networks [15]. Therefore, a novel approach to counteract reactive jamming is needed. Once reactive jamming attacks are addressed in IoT networks, other types of jamming attacks, such as constant, deceptive, and random jamming, can also be handled because reactive jamming consists of proactive jamming characteristics such as those of the other types of attacks [9].

3 System model

This section provides a reactive jamming attack model in IoT network. It shows the level of data flows, objects involved and how messages are exchanged among various objects in the network during the interaction [4, 17]. In addition, the system modeling provides a high-level of understanding of the device functionalities and describes how data flow among users and devices [5, 23]. The system model of a sender node, receiver node, and reactive jammer is illustrated in Fig. 2.

Reactive jamming is only feasible when the geometry of the system is such that the jammer's transmitted signal reaches the target receiver before it hops to a new channel or stops transmitting. As such, reactive jamming is only possible when the jammer is physically located near or between the target transmitter and receiver. If represents the fraction of each node duration that must remain not jammed for communications to succeed, then from Eq. (1), we have following inequality limiting the distances S$_2$ and S$_3$;

$$S_2 + S_3 \leq \left(\lambda R_s - R_j\right) q + S_1 \tag{1}$$

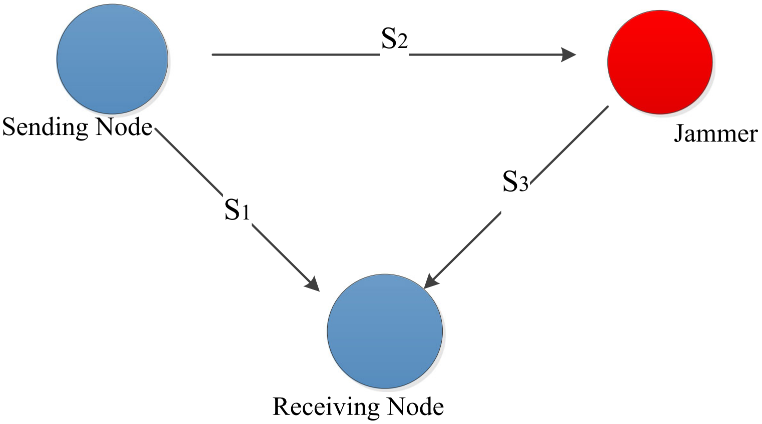


Fig. 2 System model for sender node, receiver node and reactive jammer

where R_s is the node duration, R_j is the jammer's processing time, q is the speed of light, and S_1, S_2, and S_3 are the distances indicated in Fig. 2.

The reactive jammer continuously monitors the communication medium and upon sensing a packet transmission immediately transmits a radio signal in order to cause a collision at the receiver [1, 6]. The jammer can send just enough power to corrupt a single bit to cause a received packet to fail the Cyclic Redundancy Codes (CRC) check. Normally, reactive jamming attack have the following criteria; high energy efficiency (i.e., consume low power), low probability of detection (preferably close to 0), achieve high levels of DoS (i.e., disrupt communications to the desired (or maximum possible) extent) and be resistant to PHY layer antijamming techniques (i.e., do not allow signal processing techniques to overcome the attack). Often, the criteria of interest are jamming scenario dependent [16, 31]. In other words, the jamming scenario dictates the most suitable criteria for use [35]. For instance, when malicious nodes have limited energy resources, energy efficiency will be their prime goal. Based on these criteria, different jammers tend to be effective in all cases, as long as the aforementioned criteria are possible. For example, in order to maintain a low probability of detection, the jammer can adopt techniques that are consistent with MAC layer behaviors.

4 Proposed CDCA for IoT

As mentioned, the goal of a reactive jammer is to prevent a legitimate node from sending packets by making the channel appear always busy. In this situation, a method is needed to monitor the amount of time spent waiting for the channel to be idle, as well as to check packet signal strength and node location consistency, and compare these metrics to normal traffic time to determine if jamming occurs in the channel. To perform these tasks, we introduce CDCA.

The proposed CDCA is a technique that uses a change in threshold value, as illustrated in Eq. (2), to detect and cure a network attack. The proposed technique also boosts IoT network performance (packet signal strength) and improves the location consistency checks of connected devices, thereby protecting the network against reactive jamming. The threshold value is calculated on the basis of Eq. (2).

$$\sum_{i=1}^{n} Dx_i = N \tag{2}$$

where D is MaxPacket of neighboring nodes, x_i is the distance between each node, n is the total number of nodes, and N is the threshold value.

The algorithm uses channel signal strength to determine packet consistency by checking if the data transmission value contradicts the threshold value. The node that sends the threshold value is periodically checked and its value is compared with the current value after data transmission to determine if an attack has occurred on the network. If the threshold value is greater than the MaxPacket of any particular node, then that node is considered as a jammer. However, if the threshold value is less than the MaxPacket of any particular node, then that node may or may not be considered as a jammer because it may be a result of weak signals during transmission. Nevertheless, in this case, the current value received after transmission has to be checked and compared with the sending threshold value on the communicating nodes.

When the jammer node is detected by CDCA, it notifies all other connected nodes about the jammer node to change all paths coming from itself. As a result, the defense mechanism indirectly removes the jammer node from the network. The CDCA network, attacker assumptions, and algorithm are described in the following subsections.

4.1 Network assumptions of CDCA

The following is a list of CDCA assumptions:

1. The network consists of *n* number of IoT devices that are randomly distributed.
2. All devices (nodes) have the same capability in terms of functionality. Each node serves as a gateway because every node has an IP address, and the nodes can send messages to one another.
3. All nodes can communicate with one another via single-hop or multiple-hop mode.
4. The jamming attack can be initiated by any node. A reactive jammer begins to interfere with the channel as soon as it senses an activity there.
5. Both the jamming and normal nodes are equipped with the same capability, but the jammer node is able to generate random messages (i.e., random jamming signal).

4.2 Features of CDCA

The main idea behind CDCA is to boost the IoT network performance and protect the network against reactive jamming by assigning a sending threshold value to each node in the network. The sending threshold value helps to determine a node's ability to send maximum data, which enables the proposed algorithm to detect if an attack has occurred in the network. The flow of the proposed algorithm is presented as follows:

Algorithm 1: CDCA Consistency Check

```
Required
n, MaxPacket(D): D ∈ Neighbor nodes
Input: Threshold value, $a_0$, $b_{0,}$ $a_n$, $b_{n;}$
Output: Dist, ΔThreshold value
If (MaxPacket(D) < ThresholdPacket) Then
        Signal Strength (SS) is checked for consistency
        Check the sending threshold value
        Check channel = SS consistency (SS, MaxPacket(D))
Elseif Check channel = false Then
        Jamming occurred
  Endif
If (MaxPacket(D) > ThresholdPacket) Then
        $X_0 = (a_0, b_0)$ = node_location
        $X_n = (a_n, b_n)$ = find_location (n)
        Path Analysis = dist ($X_0$, $X_n$)
        Check channel = Sent packet
Elseif Check channel = false Then
        Jamming occurred
Endif
End.
```

Where X_0 signifies the initial position of node from point a_0 to b_0, X_n signifies the actual position of node located from point a_n to b_n, *n* is the total number of nodes in the network and D signifies neighbor nodes.

The algorithm involves two phases. The first phase involves choosing the data that send the threshold value from each node and checking for the signal strength consistency of every node. As every node is able to send data after a certain interval, the algorithm can keep track of the amount of data sent by each node.

> The next phase is checking (location consistency), which is based on the sending threshold value. Every node in the network is assumed to be in one of three states: normal (i.e., non-attacker node), suspicious (i.e., suspected attacker node), and the attacker (i.e., jammer node). Initially, all nodes are in the normal state and send data to one another through single-hop or multi-hop communication mode.

Path analysis of the suspicious state node is conducted in two phases depending on the type of communication (i.e., single-hop or multi-hop) used by the nodes for exchanging data. If the suspicious source node uses the single-hop communication mode, then the algorithm can easily detect the attacker by performing single-hop path analysis. If the suspicious node uses multiple hops, the path analysis algorithm checks each hop and the packets transmitted by all nodes. If the number of packets produced by a node is more than the normal sent data, then that node is considered to be a jammer and is converted to jamming state by the algorithm. The algorithm decides that a node is in a normal state when the sending threshold packet value is the same as the MaxPacket. To determine the normal state/data, the following rules are followed:

1. If the MaxPacket is less than the ThresholdPacket, then the malicious node level is high.
2. If the MaxPacket is equal to the ThresholdPacket, then the malicious node level is normal. At this level, normal data are received.
3. If the MaxPacket is greater than the ThresholdPacket, then the malicious node level is low.

The MaxPacket denotes the maximum value of the packet from the corresponding entries of each node in the network every 1 s. Finally, the algorithm removes the jamming node by informing other nodes about it and changing the communication path in the network. The jamming node is eliminated from the network by making its path redundant and thus, the jammer node continuously emits its limited energy. Figure 3 illustrates how the algorithm functions.

4.3 Reactive jamming and detection model formulation

The reactive jamming attack and detection model are formulated as either one of two types of games: “non-cooperative” or “zero sum”. The first type aims to mitigate security issues in the IoT network. The second type is a kind of non-cooperative game between two players, one of whom is a “maximizer” and the other a “minimizer”. The maximizer strives to reach the highest level of gains, whereas the minimizer aims to keep its losses at the lowest level [1]. This approach is selected because, in the IoT network, every node (player) tends to maximize its resources (gains) to be fully active during packet transmission. These players are monitoring nodes that are responsible for detecting the reactive jamming attack. The players are set as

$$S = \{ S1, S2\} \tag{3}$$

where $S1$ is the monitoring node and $S2$ is the jammer.

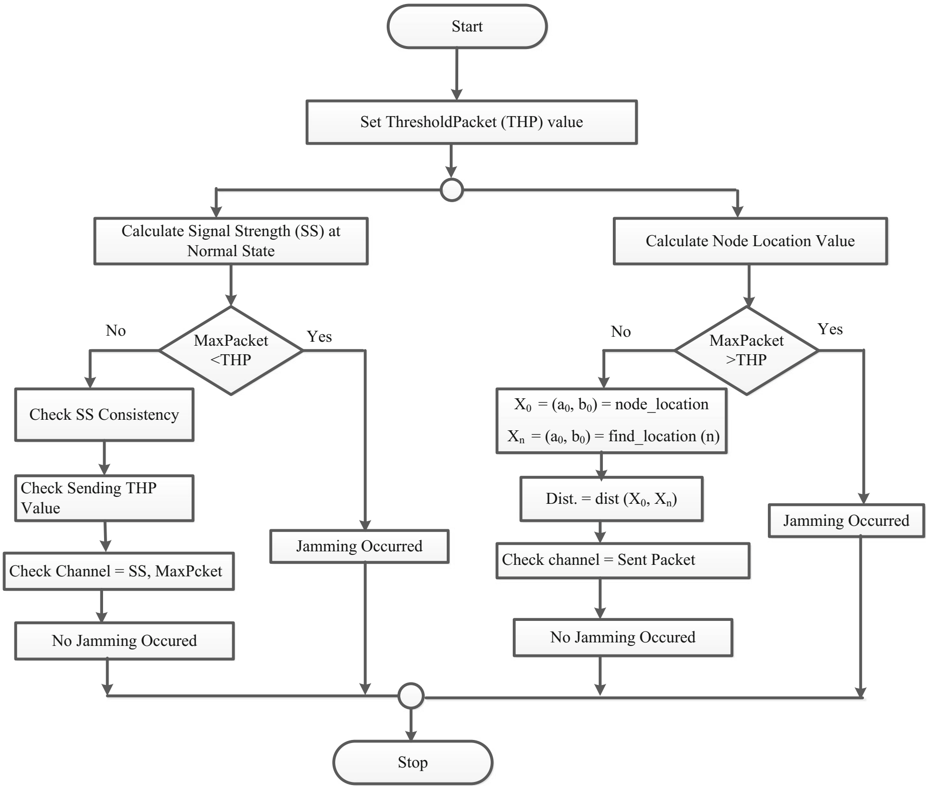

Start
Set ThresholdPacket (THP) value
Calculate Signal Strength (SS) at Normal State
Calculate Node Location Value
No
MaxPacket <THP
Yes
No
MaxPacket >THP
Yes
Check SS Consistency
Check Sending THP Value
Check Channel = SS, MaxPcket
No Jamming Occured
Jamming Occurred
X₀ = (a₀, b₀) = node_location
Xₙ = (a₀, b₀) = find_location (n)
Dist. = dist (X₀, Xₙ)
Check channel = Sent Packet
No Jamming Occured
Jamming Occurred
Stop

Fig. 3 Flowchart of CDCA

The tradeoffs in the model formulation are constant monitoring *(Mc)* and periodic monitoring *(Mp)*, which provide the nodes with an option to monitor the communication channel either continuously or at a predefined time interval. The reactive jamming attack strategy is adopted in this model and denoted as “ReJ”. The monitoring node uses two strategies *(Mc, Mp)* in monitoring the channel.

The strategy is denoted as

$$S = S1{*}S2 \tag{4}$$

$$S1 = \{Mc, Mp\} \tag{4a}$$

$$S2 = \{ReJ\} \tag{5b}$$

where $S1$ and S2 represent players 1 and 2, respectively.

Player utility functions are also considered. The utility function indicates the effectiveness of the monitoring node related to attacks that are effectively detected and those classified as false. For the monitoring nodes, detection rate and false positive rate are the two utility functions considered. The jammer utility aims to prevent successful packet transmission and reduces network throughput by introducing a DoS attack into the network.

The utility function *(U)* is given as

$$\{U\} = \{U1, U2\} \tag{6}$$

where $U1$ and $U2$ are the detection rate and attack gain, respectively.

Based on Eqs. (2), (3), and (4), a reactive jamming strategy model for an IoT network is formulated for both the *Mc* and *Mp* strategies as follows:

$$Mc = A_d\left(G_a - P_{rej}\right), \left(G_d - P_c\right) \tag{7}$$

$$Mp = A_d\left(tG_a - P_{rej}\right), t\left(A_d G_d - P_p\right) \tag{8}$$

where A_d denotes the attack duration, t represents *Mp* time, G_d denotes attack detection gain, G_a represents the gain for an attack that is launched successfully, P_{rej} is the payoff for reactive jamming, and P_c and P_p are both payoffs when *Mc* and *Mp* are used to detect the attack. Gain is when an attack is successfully detected or launched. Payoff is the cost of detecting or launching an attack.

Equations 4 and 5b express the reactive attack strategy for both *Mc* and *Mp*. The reactive jammer launches an attack only when it detects an activity in the communication channel. Attack duration A_d is defined as the period when a reactive jammer launches an attack on the channel. In the case of *Mc,* paying cost P_{rej} in period A_d enables the reactive jammer to accomplish its gain G_a. Otherwise, the jammer gains G_d by paying the cost of constant monitoring P_c. In *Mp,* the reactive jammer can achieve gain G_a at time t by paying P_{rej} in each attack duration A_d. Otherwise, the jammer gain G_d at each A_d is achieved by paying P_p after t. Table 2 (Appendix) provides definitions of all the mathematical symbols used in sections 3 and 4.

5 Implementation details

This section describes the implementation of CDCA using Cooja simulator [31] under varying traffic intervals, number of malicious nodes, a realistic condition, and random mobility.

5.1 Simulation parameter

We used the Cooja simulator to create the attacks. The simulation parameters are based on IEEE 802.15.4 radio standard. The details of the parameters are provided in Table 1.

These values provide the basis for proper evaluation of jamming attacks [30].

Table 1 Simulation Parameters

Parameters	Value
Initial Energy	100 J
Idle Power	31 mW
Receiving Power	35 mW
Sleep Power	15 µW
Transmission Power	31 mW

6 Results and discussions

In this section, the performance of the proposed algorithm is compared with that of the existing TJC countermeasure in [4] and traditional approach in [34] to detect reactive jamming.

The MAC protocol is Contiki and the routing protocol [16] is ad-hoc routing. The simulation is run 50 times and the results are compared with those of the state-of-the-art countermeasures.

The attacks are simulated under the following conditions:

1. Simulation performed under varying traffic intervals: This condition is important to measure and ascertain the performance of jamming attacks on the network and determine the countermeasure in different traffic situations. The traffic interval ranges from 1 s to 10 s, where 1 s is considered as fast and 10s is slow.
2. Under a varying number of malicious nodes: This set of simulations is used to analyze the effect of network attacks and determine the appropriate countermeasures by accumulating all malicious objects. Nodes 1, 3, 5, 7, and 15 are considered as malicious nodes. The traffic interval is set to 1 s, which is the fastest network traffic.
3. Under a realistic condition: A realistic condition is one where the nodes in the network do not transmit information at the same time, i.e., they transmit information at different intervals. The traffic interval varies randomly between 1 and 10 s.
4. Under random mobility: The set of simulations provides a realistic behavior of CDCA. In this scenario, random mobility is added to each node and the traffic interval varies between 1 and 10 s. Furthermore, mobility speed varies from 1 km/h to 25 km/h.

The performance metrics are energy consumption, traffic delay, and network throughput.

6.1 Energy consumption performance

Figure 4 demonstrates the average energy consumption and number of malicious nodes as the time interval varies. Fig. 4a indicates the superiority of the proposed CDCA approach to contemporary schemes in terms of reducing energy consumption.

The main reason is that the proposed algorithm detects reactive jammers and isolates them from the network, thereby reducing the energy consumption level that occurs as a result of reactive jamming. For example, with the traditional approach, the average energy consumption during reactive jamming from interval 1 s to 5 s is approximately 15%, and the percentage continues to increase as the interval increases. However, when the proposed algorithm is introduced into the network at the same interval, the energy consumption is reduced to

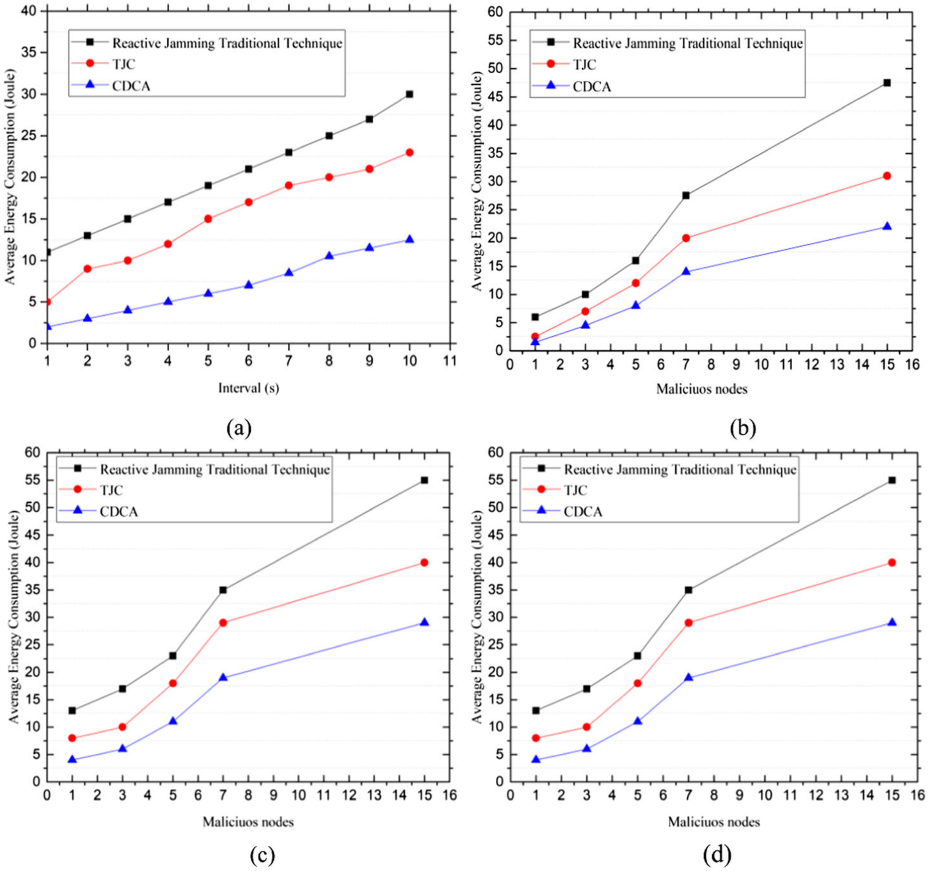


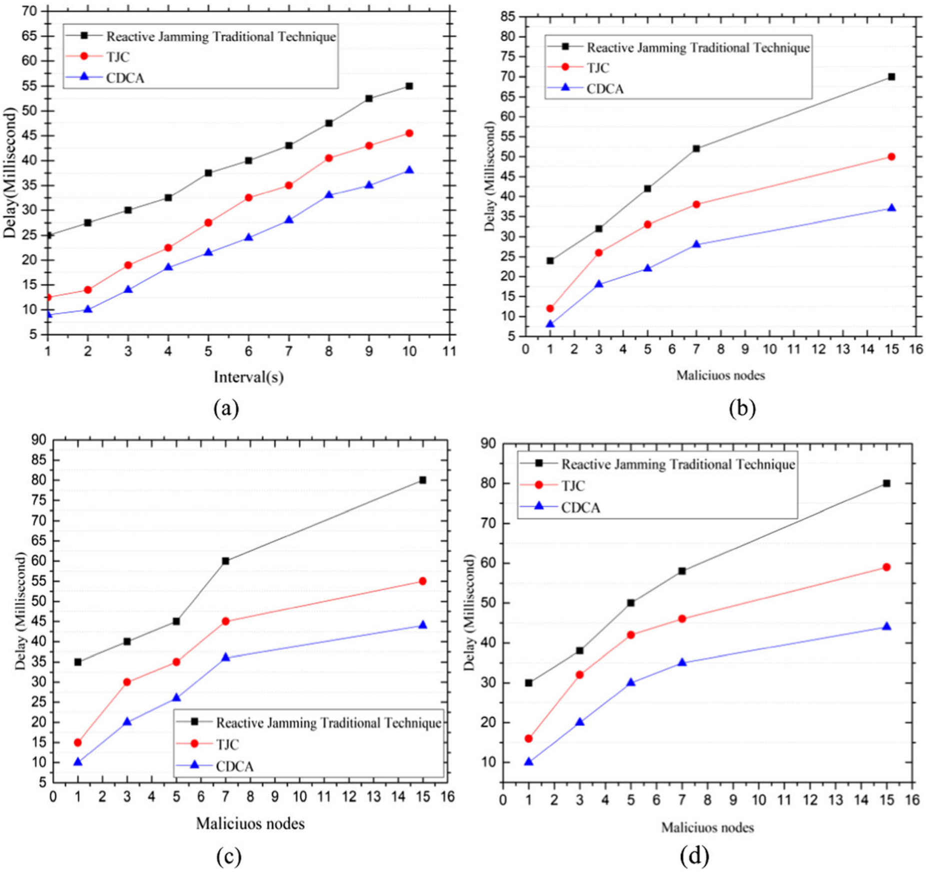
Fig. 4 Average energy consumption as a function of varying traffic interval in (a), malicious nodes in (b), malicious nodes with realistic conditions in (c), and malicious nodes with mobility in (d)

approximately 3% compared with 11% for the existing TJC approach. As expected, the energy consumption increases with the increased time interval for all the approaches. Reactive jamming has the highest energy consumption because it transmits signals only when it detects that the normal node is sending packets to other nodes. Figure 4b elucidates the average energy consumption with a varying number of malicious nodes, which are set to 1, 3, 5, 7, and 15. The point of identifying malicious nodes is to conduct a realistic and accurate analysis of CDCA when the amount of jamming increases in the network. The result shows an increase in the energy consumption level when malicious nodes are present. The energy consumption levels in a normal reactive jamming situation based on the traditional approach at malicious nodes 1 and 5 are approximately 10 and 20%, respectively, which increase to approximately 47% at malicious node 15. Although the energy consumption level drops to approximately 2 and 8% at malicious nodes 1 and 5, respectively, it is approximately 22% at malicious node 15 when the proposed algorithm is applied. This result is better than that of the existing TJC, which consumes approximately 4 and 14% at malicious nodes 1 and 3, respectively, and then up to around 33% at malicious node 15. The proposed algorithm saves energy because its detection mechanism reduces the energy consumption of the jamming node by not sending data to an active state when an attack is detected and by detecting multiple jamming attacks on a single path, thereby proving that CDCA is efficient and can effectively treat the attacks. Figure 4c shows the average energy consumption in a realistic condition with malicious nodes. The realistic condition simply means that packets are transmitted randomly in the network among nodes between intervals, i.e., each node sends packets at a different time. Simulating the realistic condition is important to understand the performance of CDCA during a reactive attack. The result indicates that, with the traditional technique, the energy consumption during normal reactive jamming at node 1 is approximately 13%, which increases to approximately 55% at node 15 in a realistic condition. However, the energy consumption level decreases to approximately 4 and 29% at nodes 1 and 15, respectively, and the network performance improves when the proposed algorithm is applied to the network in a realistic condition. Moreover, it is 4% less than that of the existing TJC at node 1 and 11% at node 15 in the same condition. Figure 4d explains the average energy consumption under mobility with malicious nodes. The simulation aims to test the flexibility of the proposed countermeasure in the IoT network. The results show that under mobility and when the traditional technique is applied, the average energy consumption of normal reactive jamming at node 1 is approximately 15%, which increases to approximately 78% at node 15 as the number of malicious nodes increase their mobility. The proposed algorithm reduces the energy consumption to approximately 3 and 23% at nodes 1 and 15, respectively, thereby improving the network performance as the number of malicious nodes increases. The energy consumption is 6 and 20% less than that of the existing TJC under mobility.

6.2 Traffic delay

Figure 5 demonstrates the traffic delay and the number of malicious nodes as the time interval varies. Figure 5a indicates the superior performance of the CDCA approach compared with its contemporaries in terms of traffic delay.

The reason is that the proposed algorithm can detect, stop, and separate/remove the jamming node from the network. In addition, isolating the jamming node indirectly eliminates traffic jams in the channel, thereby reducing the delay. The result indicates that in a normal reactive jamming condition, the traffic delay using the traditional approach at interval 1 s is approximately 25%, but when the proposed algorithm is applied to the network at the same interval, the traffic delay decreases by almost 9%, which is 3% less than that of the existing algorithm. However, the percentage continues to increase in proportion to an increase in the interval. Figure 5b illustrates the performance of CDCA with traffic delay under a varying number of malicious nodes. The results show that the traffic delay in a normal reactive jamming condition using the traditional technique is approximately 24% at malicious node 1, and the percentage of delay increases with an increase in the number of malicious nodes; for example, the delay increases to approximately 70% at malicious node 15. However, the proposed algorithm reduces the traffic delay to approximately 8 and 38% at malicious nodes 1 and 15, respectively, compared with the existing approach with approximately 12 and 50% traffic delay at the same number of malicious nodes. The proposed detection algorithm can reduce traffic delay by decreasing the channel waiting time. Figure 5c elucidates the traffic delay under a realistic condition with varying number of malicious nodes. The delay of the normal reactive jamming is approximately 35% at node 1, and this rate increases to approximately 80% at node 15 in a realistic condition when the traditional technique is used. However, when the proposed CDCA is applied, the delay is reduced to approximately 10% at node 1 and approximately 44% at node 15 in a realistic condition, which is around 5% better



(a)

(b)

(c)

(d)

Fig. 5 Effect of varying traffic interval (a), malicious nodes (b), malicious nodes with realistic conditions (c), and mobility with malicious nodes in (d) on delay

than the result of the existing TJC at node 1 and 11% better at node 15. Figure 5d shows the results of CDCA for traffic delay under mobility with malicious nodes. The result indicates that during normal reactive jamming with the traditional approach under mobility, the traffic delay is approximately 30% at node 1, which increases to around 80% at node 15, thereby degrading the network performance. When CDCA is applied, the network performance improves by approximately 10% in terms of delay compared with that of TJC at node 1 and almost 11% at node 15 under mobility. The introduction of mobility causes an increase in energy consumption and delay because additional time is needed to determine and calculate the threshold value of each node and a larger amount of energy is needed to scan the pathway to detect malicious nodes in the network.

6.3 Network throughput

Figure 6 demonstrates the network throughput and the number of malicious nodes as the time interval varies. Figure 6a shows that the network throughput degrades to approximately 42% at an interval of 1 s in a normal reactive jamming condition using the traditional technique.

However, the proposed algorithm improves the network throughput to approximately 86%, which is 10% higher than that of the existing algorithm at the same interval of 1 s.

The result shows that the network throughput performance decreases as the interval increases. The normal reactive jamming condition keeps the communication channel busy, thereby affecting the network throughput as a result of heavy pending traffic incurred on each node. Figure 6b demonstrates the result of the network throughput as the number of malicious nodes increases. The figure shows that when the traditional technique is used, the network throughput is approximately 43% at malicious node 1 and decreases to around 31% at malicious node 15 in a normal reactive jamming condition. The network throughput improves to approximately 85% at malicious node 1 and decreases to around 54% at malicious node 15 when the proposed algorithm is applied. These rates are 10 and 5% better than those of the existing TJC at malicious nodes 1 and 15, respectively, because the proposed algorithm provides rapid channel accessibility to nodes during reactive jamming. Figure 6c displays the network throughput under a realistic condition with malicious nodes. The result indicates that when the traditional technique is used, the throughput is approximately 41% at node 1 and decreases to around 26% at node 15 during normal reactive jamming under a realistic condition. The proposed algorithm improves the network throughput to approximately 72% at node 1

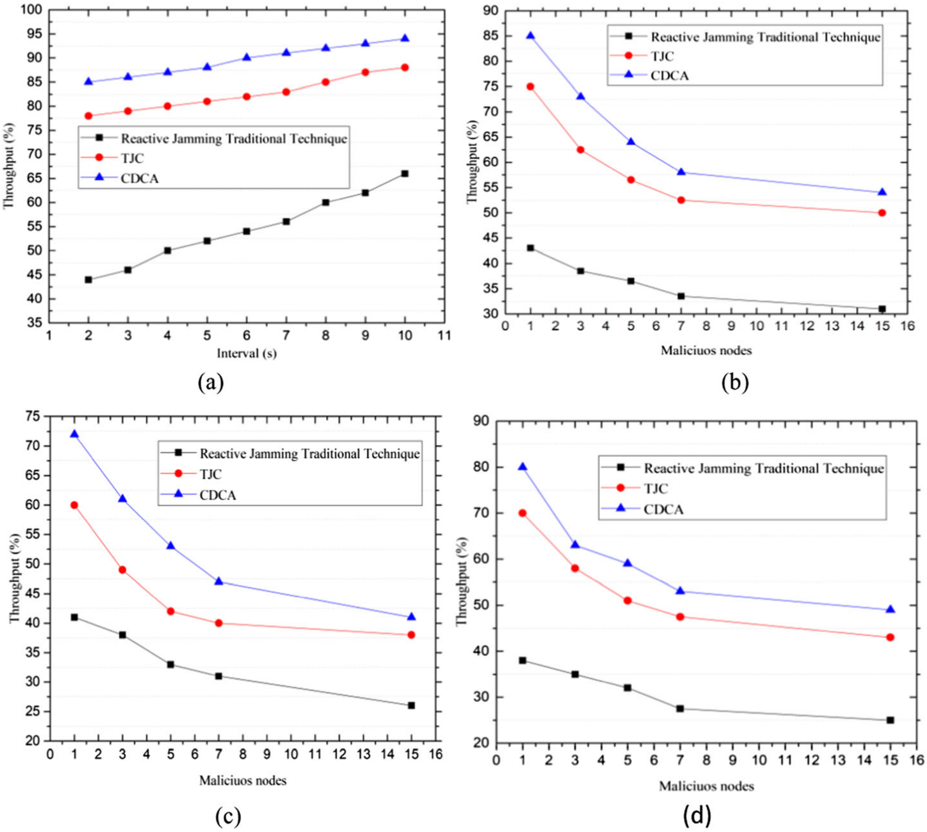


Fig. 6 (a) Throughput as a function of varying traffic interval (a), malicious nodes (b), malicious nodes with realistic conditions (c), and mobility with malicious nodes (d)

and to around 41% at node 15 in a realistic condition. This result is 12% better than that of the existing TJC at node 1 and 3% better than that of TJC at node 15 in a realistic condition. Figure 6d illustrates the network throughput comparison of CDCA with the contemporary scheme under mobility with malicious nodes. The result shows that with the traditional approach, the throughput during the normal reactive jamming condition under mobility is approximately 38% at node 1 and decreases gradually to 25% at node 15, thereby affecting the network performance. However, the network throughput with CDCA is approximately 80% under mobility at node 1 and decreases to 49% at node 15 as the number of malicious nodes increases. This result is better than the throughput of the existing method at approximately 70% at node 1 and around 43% at node 15 under mobility. The reason for the network throughput reduction is an increase in the time needed to detect the jamming node.

7 Conclusion

This paper discussed the various countermeasures against reactive jamming attacks on IoT networks. The simulation results showed that during a reactive jamming attack, the proposed CDCA performed better than the existing countermeasure under three simulated conditions in terms of energy consumption, traffic delay, and network throughput. Furthermore, compared with the state-of-the-art countermeasures, the proposed CDCA exhibited high performance under varying traffic intervals and under a number of malicious nodes. For instance, CDCA reduces the network energy consumption to approximately 3% under varying traffic intervals compared to the existing schemes that consume almost 11% under the same condition. CDCA also demonstrates a good performance in terms of energy consumption in the presence of malicious nodes compared to the contemporary schemes by reducing the amount of energy consumed at malicious node 1 and 3 to around 2% and 8%, respectively. In addition, CDCA shows a superior performance in terms of delay under varying traffic interval by reducing the traffic delay to around 9%, which is 3% less than the existing schemes. In the presence of malicious nodes, it also reduces the delay at malicious node 1 and 15 to approximately 8% and 38%, respectively, which is better the existing schemes that have a delay of around 12% and 50%, respectively. Finally, CDCA improves the network throughput up to 86%, which is 10% better than that of the existing scheme. The main benefit of the novel CDCA is that its defense mechanism supports an increase in the number of jamming nodes in the network. Moreover, it uses the change in threshold value, signal strength and checks packet consistency to easily detect and treat an attack within the network. In the future, we are planning to extend this study by including other types of jamming attacks on IoT networks.

Acknowledgments The work is supported by Geran Bantuan Khas Penyelidikan (BKS084-2017) and Deanship of Scientific Research at King Saud University through Research Group Project NO. (RG-1435-051).

Appendix

Table 2 List of Mathematical Notations

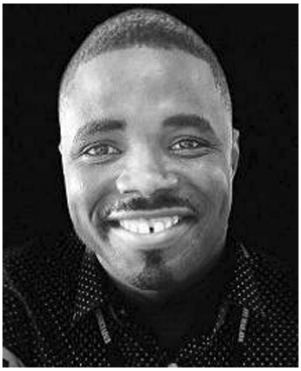
Parameter	Description
	Fraction of each node duration
R_s	Node duration
R_j	Jammer's processing time
q	Speed of Light
S_1, S_2, S_3	Distances
D	MaxPacket of neighboring nodes
x_i	Distance between each node
n	Total number of nodes
N	Threshold value
X_0	Initial position of nodes from point a_0 to b_0
X_n	Final position of nodes from point a_n to b_n
S1	Monitoring node
S2	Jammer
M_c	Constant monitoring
M_p	Periodic monitoring
U	Utility Function
A_d	Attack duration
t	M_p time
G_d	Attack detection gain
P_{rej}	Payoff for reactive jamming
P_i	Payoff for M_c
P_P	Payoff for M_c

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Abdalzaher MS, Seddik K, Elsabrouty M, Muta O, Furukawa H, Abdel-Rahman A (2016) Game theory meets wireless sensor networks security requirements and threats mitigation: A survey. Sensors (Switzerland) 16(7):22–27
2. Aman W, Snekkenes E (2015) Managing Security trade-offs in the internet of things using adaptive security. In Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for (pp. 362-368). IEEE
3. Babar SD, Prasad NR, Prasad R (2013) Jamming attack: Behavioral modelling and analysis. In Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2013 3rd International Conference on (pp. 1-5). IEEE
4. Babar S, Stango A, Prasad N, Sen J, Prasad R (2011) Proposed embedded security framework for internet of things (iot). In Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on (pp. 1-5). IEEE
5. Derhab A, Bouras A, Senouci MR, Imran M (2014) Fortifying intrusion detection systems in dynamic Ad Hoc and wireless sensor networks. Int J Distrib Sens Netw 10(12):608162
6. Ergul E (2017) Relative attribute based incremental learning for image recognition. CAAI Trans Intell Technol 2(1):1–11
7. Fadele AA, Othman M, Hashem IAT, Alotaibi F (2017) Internet of things Security: A Survey. J Netw Comput Appl 88:10–28
8. Fu H, Chen H, Zhu Y, Yu W (2017) FARMS : Efficient mapreduce speculation for failure recovery in short jobs. Parallel Comput 61:68–82

9. Ganeshkumar P, Vijayakumar KP, Anandaraj M (2016) A novel jammer detection framework for cluster-based wireless sensor networks. EURASIP J Wirel Commun Netw 2016(1):35
10. Goyal D, Tripathy MR (2012) Routing protocols in wireless sensor networks: A survey. In 2012 Second International Conference on Advanced Computing & Communication Technologies (pp. 474-480). IEEE
11. Hatzivasilis G, Papaefstathiou I, Manifavas C (2017) SCOTRES: Secure Routing for IoT and CPS. IEEE Internet Things J 73100(c):1–1
12. Heo J, Kim J, Bahk S (2017) Dodge-Jam : Anti-Jamming Technique for Low-power and Lossy Wireless Networks. IEEE Access 2(3):345–359
13. Huang K, Zhang Q, Zhou C, Xiong N, Qin Y (2017) An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning. IEEE Trans Syst Man Cybern Part A Syst Hum 47(10): 2704–2713
14. Kasinathan P, Pastrone C, Spirito MA, Vinkovits M (2013) Denial-of-Service detection in 6LoWPAN based Internet of Things. In 2013 IEEE 9th international conference on wireless and mobile computing, networking and communications (WiMob) (pp. 600-607). IEEE
15. Lee JJ, Lim J (2012) Effective and efficient jamming based on routing in wireless ad hoc networks. IEEE Commun Lett 16(11):1903–1906
16. Li S, Tryfonas T, Li H (2016) The Internet of Things: a security point of view. Internet Res 26(2):337–359
17. Liu H, Hu L, Ma L (2017) Online RGB-D person re-identification based on metric model update. CAAI Trans Intell Technol 2(1):48–55
18. Lu C (2014) Overview of Security and Privacy Issues in the Internet of Things. 1–11
19. Lu Y, Wu S, Fang Z, Xiong N, Yoon S, Park DS (2017) Exploring finger vein based personal authentication for secure IoT. Futur Gener Comput Syst 77:149–160
20. Mpitziopoulos A, Gavalas D, Konstantopoulos C, Pantziou G (2014) A survey on jamming attacks and countermeasures in WSNs. IEEE Commun Surv Tutorials 11(4):42–56
21. Nolin J, Olson N (2016) The Internet of Things and convenience. Internet Res 26(2):360–376
22. Pei E, Han H, Sun Z, Shen B, Zhang T (2015) LEAUCH: low-energy adaptive uneven clustering hierarchy for cognitive radio sensor network. EURASIP J Wirel Commun Netw 2015(1):122
23. Qiu Y, Ma M (2016) A Mutual Authentication and Key Establishment Scheme for M2M Communication in 6LoWPAN. IEEE Trans Ind Informatics 12(6):2074–2085
24. Raymond DR, Marchany RC, Brownfield MI, Midkiff SF (2011) Effects of denial-of-sleep attacks on wireless sensor network MAC protocols. IEEE Trans Veh Technol 58(1):367–380
25. Rehab I, Tian X, Gu F, Ball AD (2018) The influence of rolling bearing clearances on diagnostic signatures based on a numerical simulation and experimental evaluation. Int J Hydromechatronics 1(1):16–46
26. Restuccia F, D'Oro S, Melodia T (2018) Securing the Internet of Things: New Perspectives and Research Challenges. arXiv preprint arXiv:1803.05022
27. Sankaran S (2016) Lightweight security framework for IoTs using identity based cryptography. In Advances in Computing, Communications and Informatics (ICACCI), 2016 International Conference on (pp. 880-886). IEEE
28. Strasser M, Danev B, Čapkun S (2010) Detection of reactive jamming in sensor networks. ACM Trans Sens Networks 7(2):1–29
29. Suo H, Wan J, Zou C, Liu J (2012) Security in the internet of things: A review. Proc - 2012 Int Conf Comput Sci Electron Eng ICCSEE 2012 3:648–651
30. Tarkowski M, Rzymowski M, Kulas L, Nyka K (2017) Improved jamming resistance using electronically steerable parasitic antenna radiator. 17th IEEE Int Conf Smart Technol EUROCON 2017 - Conf. Proc, pp 6–8
31. Romdhani I, Qasem M, Al-Dubai AY, Ghaleb B (2016) Cooja simulator manual. Edinburgh Napier University
32. Weber SG, Martucci LA, Ries S, Mühlhäuser M (2010) Towards trustworthy identity and access management for the future internet. In Proc. 4th International Workshop on Trustworthy Internet of People, Things & Services (IoPTS) (Vol. 29)
33. Wood AD, Stankovic JA, Zhou G (2011) DEEJAM : Defeating Energy-Efficient Networks. IEEE Sensors J 2(3):60–69
34. Xu W, Ma K, Trappe W, Zhang Y (2013) Jamming sensor networks: Attack and defense strategies. IEEE Netw 20(3):41–47
35. Yang X, Xie L, Han J, Wang Z (2017) Cognitive-affective regulation process for micro-expressions based on Gaussian cloud distribution. CAAI Trans Intell Technol 2(1):56–61
36. Yaqoob I et al (2017) The rise of ransomware and emerging security challenges in the Internet of Things. Comput Netw 129:444–458
37. Zhang W, Zhu S, Tang J, Xiong N (2018) A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks. J Supercomput 74(4):1779–1801
38. Zou Y, Zhu J, Wang X, Hanzo L (2016) A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends. Proc IEEE 104(9):1727–1765



Alaba Ayotunde Fadele received his B.Sc. degree in Computer Science in 2008 from the Nasarawa State University, Keffi, Nigeria. He obtained his master degree in Computer Science from the Ahmadu Bello University, Zaria, Nigeria. He is currently a PhD candidate in the University Malaya, Malaysia. He has published several papers in top conferences and reputable journals. His current research interests include big data Analytics and IoT.

Mazliza Othman is an Associate Professor with the Faculty of Computer Science and IT at the University of Malaya, Malaysia. She received her B.Sc. degree in Computer Science from Universiti Kebangsaan Malaysia. Later she obtained a M.Sc. degree in Data Communication Networks and Distributed Systems and Ph.D. degree in mobile computing from the University College London, UK. Her research interests include pervasive computing and self-organizing net- works. She is the author of ‘Principle of Mobile Computing and Communications’.

Ibrahim Abaker Targio Hashem is working as a faculty member in Taylor's University, Malaysia. He received his Ph.D. (computer science) from the University of Malaya in 2017. He received his M.S. degree in computing in 2012 and his B.E. degree in computer science in 2007. He obtained professional certificates from CISCO (CCNP, CCNA, and CCNA Security) and APMG Group (PRINCE2 Foundation, ITIL v3 Foundation, and OBASHI Foundation). His main research interests include big data, cloud computing, distributed computing, and IoT, Security.

Ibrar Yaqoob received his Ph.D. (Computer Sciences) degree from the University of Malaya, Malaysia, in 2017. He won BrightSparks scholarship for his Ph.D. studies that is considered prestigious in Malaysia. Prior to that, he received his BS. IT (Hons.) degree from the University of the Punjab, Gujranwala Campus, Pakistan, in 2012. He worked as a researcher at the Centre for Mobile Cloud Computing Research (C4MCCR), University of Malaya. His research has appeared in several prestigious journals, such as IEEE Wireless Communications, IEEE Communications Magazine, Information Systems, Computer Networks, to name a few. His numerous research articles are very famous and among the most downloaded in top journals. One of his journal articles on big data topic is Malaysia's number one most-cited article. He has reviewed more than 100 articles for the top ISI-Indexed journals and conferences. He has served as a technical program committee member for various international conferences, which were held in Spain, Portugal, United Kingdom, Germany, and China. He is also serving as an associate editor and a track chair in the IEEE Access Journal and IEEE HealthCom 2017, respectively. His research interests include big data, edge computing, mobile cloud computing, the Internet of Things, Computer networks, and Software-Defined Networks.

Muhammad Imran is working as Assistant Professor in the College of Computer and Information Sciences, King Saud University (KSU) since 2011. He worked as a Postdoctoral Associate on joint research projects between KSU, University of Sydney and Iowa State University. He is a Visiting Scientist with Iowa State University, USA. His research interest includes mobile ad hoc and sensor networks, WBANs, M2M, IoT and fault tolerant computing. He has published more than 80 publications in refereed international conferences and journals. His research is financially supported by several grants. Recently, European Alliance for Innovation (EAI) has appointed him as a Co-Editor in Chief for EAI Transactions on Pervasive Health and Technology. He also serves as an associate editor for Wireless Communication and Mobile Computing Journal (Wiley), IET Wireless Sensor Systems, International Journal of Autonomous and Adaptive Communication Systems (Inderscience) and International Journal of Information Technology and Electrical Engineering. He served/serving as a guest editor for IEEE Communications Magazine, Computer Networks (Elsevier), International Journal of Autonomous and Adaptive Communications Systems and International Journal of Distributed Sensor Networks. He has been involved in a number of conferences and workshops in various capacities such as a chair, co-chair and technical program committee member. These include IEEE ICC, Globecom, AINA, LCN, IWCMC, IFIP WWIC and BWCCA. He has received a number of awards such as Asia Pacific Advanced Network fellowship.

Muhammad Shoaib received his Ph.D. degree in Communication and Information System from Beijing University of Posts and Telecommunications, China (2010). He received his M. Eng. (2005) and B.Eng. (1995) from NED University of Engineering and Technology, Karachi. His areas of research include Video Compression techniques, Multilayer video coding, Wireless Networks, and Information Security. He has published a number of research articles in International conferences and journals. He worked as a Senior Manager (IP Operations, South) in Pakistan Currently, he is working as an Assistant Professor in the College of Computer and Information Sciences at King Saud University.

Affiliations

Alaba Ayotunde Fadele[1] • Mazliza Othman[1] • Ibrahim Abaker Targio Hashem[2] • Ibrar Yaqoob[3] • Muhammad Imran[4] • Muhammad Shoaib[4]

Ibrar Yaqoob
ibraryaqoob@ieee.org

Muhammad Imran
dr.m.imran@ieee.org

Muhammad Shoaib
muhshoaib@ksu.edu.sa

[1] Faculty of Computer Science and information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

[2] School of Computing and IT, Taylors University, Subang Jaya, 47500 Selangor, Malaysia

[3] Department of Computer Science and Engineering, Kyung Hee University, Yongin 446-701, South Korea

[4] College of Applied Computer Science, King Saud University, Almuzahmiyah 11451, Saudi Arabia