A Gaming and Trust-Model-Based Countermeasure for DIS Attack on 6TiSCH IoT Networks

Alakesh Kalita, *Member, IEEE*, Mohan Gurusamy, *Senior Member, IEEE*, and Manas Khatua, *Member, IEEE*

***Abstract*—The 6TiSCH communication architecture provides delay-bounded packet delivery, energy efficient, and reliable data-delivering communication in mission-critical Internet of Things (IoT) applications. It uses IETF's 6TiSCH minimal configuration (6TiSCH-MC) standard for resource allocation during network formation and routing using a routing protocol for low power and lossy network (RPL) as routing protocol. In RPL, the DODAG information solicitation (DIS) control packet is used to solicit routing information from the existing networks. However, it is observed that malicious transmission of this DIS packet can severely affect the 6TiSCH networks in terms of nodes' network joining time and energy consumption. Therefore, designing countermeasures of *DIS attack* in 6TiSCH network has become critically important. Additionally, the existing works neither considered all the possible parameters together for detecting DIS attack nor energy efficient, and create control packet overhead. In this work, we model noncooperative gaming to determine the optimal probability of responding to a DIS packet. Subsequently, we design a trust model to detect malicious DIS transmission in 6TiSCH networks. Finally, we merge both the proposed gaming model and trust model to propose a scheme—gaming and trust-based countermeasure (GTCM) to reduce the effect of *DIS attack* in 6TiSCH networks. We implement the GTCM on Contiki-NG and validate it using open source FIT IoT-LAB testbed. Our experimental testbed results show that GTCM reduces the effect of *DIS attack* in terms of pledges' (new nodes) joining time and energy consumption significantly.**

***Index Terms*—6TiSCH, game theory, Internet of Things (IoT), routing protocol for low power and lossy network (RPL) attack, time slotted channel hopping (TSCH), trust model.**

I. Introduction

THE IPV6 over the time-slotted channel hopping (TSCH) mode of IEEE 802.15.4e [1] (6TiSCH) wireless communication provides reliable, time bounded, and energy-efficient communication in large-scale multihop Internet of Things (IoT) applications [2], [3]. The 6TiSCH layer of the 6TiSCH protocol stack provides interoperability between the TSCH MAC behavior (Layer-2) and IETF's upper layer protocol stack. The 6TiSCH layer uses the IETF's 6TiSCH minimal configuration (6TiSCH-MC) standard [4] for allocating resource during the formation of 6TiSCH networks and it uses routing protocol for low power and lossy network (RPL) [5] as its *de-facto* routing protocol. In brief, RPL constructs a loop-free destination-oriented directed acyclic graph (DODAG) routing tree for upward and downward routing in 6TiSCH networks. For forming/constructing the DODAG, RPL uses a dedicated control packet, i.e., DODAG information object (DIO). The new joining nodes (aka pledge) require this DIO packet to join the DODAG. However, before getting the DIO packet, the pledge should receive another control frame named enhanced beacon (EB) to get synchronized with the underlying TSCH network. EB contains the basic network information, whereas DIO contains the routing information. Hence, the pledges require both control the packets to join 6TiSCH networks.

The devices used in low power and lossy networks (LLNs), such as 6TiSCH are resource constrained in terms of processing capacity, memory, and energy. Hence, the 6TiSCH devices should efficiently consume their energy in order to achieve a longer network lifetime. However, many researchers have shown that RPL is exposed to various security threats, such as *sinkhole attack*, *hello flooding attack* or *DIS attack*, *RPL version attack*, *packet forwarding attack*, *sybil attack*, and *wormhole attacks* [6], [7], [8], [9]. These attacks severely degrade the performance of RPL-based LLNs by increasing the energy consumption of the nodes, rerouting the packets via the wrong path, and congesting the network. DODAG information solicitation (DIS) *attack* is one type of such RPL attack (aka hello flooding attack), which increases the energy consumption of the legitimate nodes and exhausts the network capacity. In an RPL-based 6TiSCH network, a node transmits a multicast DIS control packet on several occasions, such as when it does not receive DIO packet for more than some predefined configurable period, current parent is not fresh, or during routing inconsistency. The nodes that have already joined the network (hereafter, we call them joined nodes) transmit several DIO packets quickly in response to the DIS packets. The malicious nodes exploit this RPL setting to increase the congestion in the 6TiSCH networks by the flooding of DIS packets and, consequently, by the DIO packets. Please note that in 6TiSCH networks, all control packets, such as EB, DIO, DIS, and

Manuscript received 29 August 2022; revised 6 December 2022; accepted 1 January 2023. Date of publication 5 January 2023; date of current version 23 May 2023. This work was supported in part by the National University of Singapore through MoE AcRF Tier-2 Grant and through NUS WBS under Grant A00094520000 (MoE T2EP50120-0021). *(Corresponding author: Alakesh Kalita.)*

Alakesh Kalita and Mohan Gurusamy are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (e-mail: alakesh.kalita1025@gmail.com; gmohan@nus.edu.sg).

Manas Khatua is with the Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, Guwahati 781039, India (e-mail: manaskhatua@iitg.ac.in).

Digital Object Identifier 10.1109/JIOT.2023.3234300

TABLE I
LIST OF ACRONYMS

Abbreviation	Meaning
TSCH	Time Slotted Channel Hopping
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e
LLN	Low Power and Lossy network
RPL	Routing Protocol for Low-power & Lossy Networks
DODAG	Destination Oriented Directed Acyclic Graphs
EB	Enhanced Beacon
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DAO	Destination Advertisement Object

keep-alive are transmitted in the single *shared cell* of a *slotframe*.[1] In Table I, we summarized the frequently used abbreviated words.

It is noteworthy that the effect of DIS attack on 6TiSCH network has not been studied in the literature except the work in [10]. In this recent work [10], authors have shown that DIS attack can significantly increase the 6TiSCH joining time of the pledges (new nodes) and their energy consumption by performing testbed experiments. Although the works in [11], [12], and [13] studied the DIS attack and proposed various schemes to reduce its impact, these schemes considered the previous version of IEEE 802.15.4 [14], where the control packets are transmitted at any given time. On the other hand, in a 6TiSCH network, all types of control packets are transmitted by all the nodes in the limited number of shared cells. Furthermore, the previous solutions either increased the control packet overhead or were not able to detect DIS attack efficiently, so reduce its effect. Please note that several works, such as [15], [16], and [17] have shown that the 6TiSCH networks suffer from insufficient resource allocation, i.e., the number of shared cells per slotframe during their formation. This insufficient resource allocation increases the formation time of the 6TiSCH networks as well as nodes' energy consumption. Hence, it becomes obvious that a DIS attack can further degrade the performance of 6TiSCH networks in terms of pledges' joining time and energy consumption as it increases the congestion in a shared cell [10]. However, in the literature, no work has been found to reduce the impact of DIS attack on the 6TiSCH network. This motivates us to design a new scheme to deal with the DIS attack in the 6TiSCH network. Furthermore, when the malicious nodes perform DIS attack using the fictitious address, all the receiving nodes will believe that new nodes want to join the network, and transmit several DIO packets rapidly in the network. It is called Sybil DIS attack in RPL, which is also to be investigated in the context of 6TiSCH networks.

In 6TiSCH network, the joined nodes selfishly transmit their control packets in shared cells. By imposing the DIS attack in 6TiSCH networks, the malicious nodes can increase the congestion and collision in the shared cell as the joined nodes transmit their DIO packets selfishly without bothering about the congestion and collision in the shared cell. This results in increased joining time of the pledges (and so, higher 6TiSCH network formation time) and energy consumption of the nodes. So, to deal with this problem, we formulate a *noncooperative gaming model* considering the already joined nodes as *players*. This proposed gaming model is further solved using *Lagrange multiplier* and Karush–Kuhn–Tucker (KKT) *conditions* for determining the optimal probability to respond to a DIS request packet so that the network would not get congested when there is a DIS attack. Thus, the proposed gaming model is designed so that the DIS attack fails to increase the 6TiSCH network formation time and nodes' energy consumption. We further propose a *trust model* for the joined nodes to distinguish between maliciously transmitted DIS packets and regular DIS packets by using different characteristics of the DIS transmitting node and DIS request itself. Finally, we combine both the models, i.e., the gaming model and trust model, and propose a scheme gaming and trust-based countermeasure (GTCM) to deal with the DIS attack in 6TiSCH networks. In brief, the following are the core contributions of this work.

1) We design a theoretical model to analyze the impact of the DIS attack on 6TiSCH network.
2) We design a *noncooperative gaming* model to determine the optimal response probability to a DIS packet. We prove that the proposed gaming model has a unique Nash equilibrium (NE) point and provide the solution for the proposed game.
3) We further propose a *trust model* to find maliciously transmitted DIS packets in the network and propose a GTCM scheme by combining the trust model with the gaming model.
4) We implement GTCM on open source Contiki-NG [18] and evaluated using FIT IoT-LAB [19] testbed.

The remainder of this article is organized as follows. Section III summarized the existing works related to the DIS attack. In Section IV, we provide the theoretical and simulation analysis of the DIS attack on the 6TiSCH network. Section V describes the formulation of the proposed gaming model, and subsequently, in Section VI, we describe the proposed trust model. Section VII describes our proposed gaming and trust-based countermeasure scheme. Finally, in Section VIII, we provide the testbed evaluation of the proposed scheme and conclude this work in Section IX.

II. OVERVIEW OF RPL PROTOCOL AND DIS ATTACK

RPL [5] uses distance vector routing protocol to provide routing facility in LLNs such as 6TiSCH. It organizes the physical LLNs into their logical representation by constructing one or more loop-free and tree-like topologies known as DODAGs. However, every DODAG has only one root, known as border router "6BR," which provides interoperability between the Internet and IEEE 802.15.4e-based LLNs. A DODAG describes the paths from the leaf nodes (i.e., sensor nodes) to the root node and vice versa. RPL maintains its DODAG and enables auto-configuration, self-organizing, and self-healing (i.e., through global and local repair) mechanisms when there is inconsistency in the networks by using four

[1] Slotframe is a collection of several timeslots. The nodes transmit their data packets in their dedicated timeslot and control packets in shared timeslot. The physical channel and the timeslot represent a *cell*.

types of control packets, such as DIS, DIO, destination advertisement object (DAO), DAO acknowledgment (DAO-ACK). However, only the DIO packet contains the information to build the DODAG. RPL uses the Trickle algorithm [20] to regulate the transmission of DIO control packets depending on the current network stability to reduce the nodes' energy consumption and efficient bandwidth utilization. In brief, the Trickle algorithm limits the transmission of DIO packets in stable networks.

The needy nodes solicit the DIO packet by transmitting either the unicast or multicast DIS packet in the network. When the joined nodes receive a multicast DIS packet, they reset the Trickle algorithm and rapidly transmit DIO packet in the network. The joined nodes think there is an inconsistency in the network, such as a change in the routing information, which needs to be updated quickly, or a new node wants to join the network. In brief, the transmission of multicast DIS packets initiates burst transmission of DIO packets in RPL-based 6TiSCH networks so that the intended receiver of the DIO packet can receive the updated routing information in less time. Please note that each joined node transmits only one unicast DIO packet to the sender of the transmitted unicast DIS packet without resetting its Trickle algorithm. However, the malicious nodes exploit this RPL setting to increase the congestion in the 6TiSCH networks by frequently transmitting multicast DIS packets and, thus, impaling the joined nodes to transmit DIO packets repeatedly.

Even though the RPL specification mentioned several security features, their implementation is kept as "optional" (partially or fully). It is also noteworthy that the present implementations of RPL by the Contiki OS and Tiny OS have not implemented the security features mentioned in the RPL specification and so they use the unsecured mode of RPL. Furthermore, both the link layer and RPL security mechanism require some control packet exchange which significantly increases the RPL's control packet overhead, energy consumption of the nodes, and network formation time [21]. Furthermore, it is also feasible to gain access to confidential data like preinstalled encryption keys of the legitimate nodes and reprogram them as malicious nodes. Recently, the IETF introduced some strategies using flags and response spreading to deal with the DIS attack [22]. However, up to this moment, these modifications have not been practically tested and investigated so far, and the work is still in its draft stage of standardization.

III. Related Works

It is observed that due to the unavailability of standard-specific security models for resource-constrained IoT devices, IoT networks become an easy target for various security attacks. Apart from the information security requirements, such as confidentiality, integrity, availability, authenticity, and nonrepudiation, IoT networks also require security in access control and authorization and protection against Denial-of-Service (Dos) attacks [23], [24], etc. Even though many advanced techniques, such as machine learning, deep learning, and Blockchain-based approaches have been proposed [25], [26] to deal with different attacks in IoT networks, it is not always possible to use these approaches in resource-constrained IoT devices because of their limited processing capacity, memory, and power supply.

In this work, we mainly consider the DIS effect in 6TiSCH networks. The main goal of a *DIS attack* is to create burst transmission of DIO packets in 6TiSCH, which increases the energy consumption of both transmitter and receiver nodes. It is further observed that DIS attack severely congests the shared cell in 6TiSCH networks, which increases the collision, and so degrades the performance of the 6TiSCH network in terms of its formation time and energy consumption. Although several works exist in the literature on different RPL's vulnerabilities and their countermeasures, only a few considered the DIS attack on RPL-based LLNs and proposed countermeasures. In this section, we discuss the works related to DIS attack.

The work [10] is the only work that considered the DIS attack in the 6TiSCH network. By performing testbed experiments, it showed that DIS attack severely degrades the performance of 6TiSCH network in terms of nodes' joining time, energy consumption, and network stability. However, the work in [10] is limited to showing the impact of the DIS attack on the 6TiSCH network only. It did not provide any countermeasure to reduce the effect of the DIS attack on the 6TiSCH network.

The work in [11] showed how the legitimate nodes suffer from Dos in the presence of DIS attack by conducting extensive simulation experiments. The authors showed that DIS attack can severely decrease the network lifetime by increasing the energy consumption of the nodes. However, it is worth mentioning that the work in [11] considered the previous version of IEEE 802.15.4 [14], where control packets are not transmitted using shared cells, unlike the 6TiSCH networks. Furthermore, that work also did not mention any countermeasure for the DIS attack. Similarly, the authors of the work [27] also analyzed the impact of the DIS attack in terms of nodes' energy consumption and DODAG formation time by performing simulation experiments but did not provide any solution.

On the other hand, Farzaneh et al. [12] developed a distributed intrusion and detection system (IDS) based on some threshold values for detecting different RPL attacks, including DIS attacks. In that IDS, all the nodes count the number of transmitted DIS packets by their neighboring nodes and send them to the IDS. The IDS compares the counts with a fixed threshold value to detect DIS attacks in the network. However, that work is also limited to detecting the attacks only. The authors did not provide any countermeasure for any detected attack. Furthermore, this detection algorithm may fail in different network conditions because it uses fixed/static threshold values. Ioulianou and Vassilakis [28] proposed a hybrid threshold-based IDS that also uses the DIS transmitting rate of the nodes to detect DIS attacks. However, it also provided only the mechanism for detecting DIS attacks. It did not mention any countermeasure for the DIS attack. Furthermore, that work imposes communication overhead in the network while detecting DIS attacks.

On the other hand, Verma and Ranga proposed [13] Secure-RPL to detect and mitigate DIS attacks in RPL-based LLNs considering the previous version of IEEE 802.15.4. However, their detection algorithm would fail, and so does the mitigation algorithm, when the malicious nodes transmit DIS packets with fictitious addresses (i.e., fake *extended unique identifier address*, `EUI64`). As the detection algorithms depend on nodes' unique `EUI64` addresses like the work in [12], [29], and [30], that solution would not work effectively when malicious nodes change their addresses every time. In the recent work DISAM [31], the authors studied the impact of DIS attack in Blockchain-enabled RPL-based IoT network and provided a mitigation scheme. This work [31] also maintained a table like Secure-RPL [13] to store the DIS packet transmission information and based on a predefined threshold value, it detects the DIS attack and discards the DIS request. This scheme would not work efficiently if the malicious nodes transmit DAO packets after receiving DIO packets. The work RPL-MRC [32] proposed an approach to deal with the DIS attack and evaluated RPL-MRC on different network scenarios, such as varied DIS attacking rate, number of attackers, and data transmission rate. RPL-MRC reduces the number of responses of multicast DIS packet, i.e., DIO packet, by dynamically varying the next DIO transmission time. The authors use few static network parameters to tune the DIS transmission rate. Furthermore, in that work, the authors considered the previous version of IEEE 802.15.4; hence, the solution may not work efficiently in 6TiSCH networks as all the nodes transmit their control packets in a shared cell. Furthermore, this work delays the DIS transmission. So, congestion would be persisted in a shared cell of 6TiSCH networks. The authors of the works in [29] and [30] studied the DIS attack in the presence of malicious nodes which continuously change their `EUI64` addresses and proposed Bloom filter and GINI countermeasure-based solutions, respectively. However, both approaches create communication overhead in the network and also require additional storage and processing.

In brief, none of the existing work proposed any solution to reduce the effect of DIS attacks on 6TiSCH networks. Hence, in this work, we propose a scheme to reduce the impact of DIS attack on 6TiSCH network without any communication overhead, additional storage, and processing. Furthermore, our proposed scheme can also detect DIS attack with a fictitious address, i.e., can detect Sybil DIS attack and considers congestion in a shared cell of 6TiSCH networks and nodes' DIO transmitting priority for efficient transmission of DIO packet in the networks while providing countermeasure for DIS attack.

In Table II, we summarize the important features of existing works related to DIS attack and compare them with our proposed scheme.

IV. Analysis of DIS Attack

In this section, we analyze the impact of a DIS attack on 6TiSCH network by designing a semi Markov process (SMP)-based theoretical model. We further implement the DIS attack on the Contiki-NG operating system and perform simulation experiments on the Contiki-based Cooja Simulator. The frequently used symbols used in our analytical model are mentioned in Table III.

TABLE II
Existing Works Related To DIS Attack

Work	Detection mechanism		Counter measure based on				Considered 6TiSCH Network?	Evaluation technique
	Fictitious address	DIO rate	DIO rate	Fictitious address	Congestion in cell	Node's status		
[10]	✗	✓	✗	✗	✗	✗	✓	Testbed
[11]	✗	✓	✗	✗	✗	✗	✗	Simulation
[27]	✗	✓	✗	✗	✗	✗	✗	Simulation
[12]	✓	✓	✗	✗	✗	✗	✗	Simulation
[28]	✓	✓	✗	✗	✗	✗	✗	Simulation
[13]	✗	✓	✓	✗	✗	✗	✗	Simulation
[32]	✗	✓	✓	✗	✗	✗	✗	Simulation
[31]	✓	✗	✗	✓	✗	✗	✗	Simulation
This work	✓	✓	✓	✓	✓	✓	✓	Testbed

TABLE III
Notation

Symbol	Meaning
n	Number of nodes
N_c	Total number of channels used
L	Length of slorframe
I_{eb}	EB generation interval
p_{eb}	EB generation probability in a shared cell
P_{dio}	DIO generation probability in a shared cell
p_l	Packet loss probability
p_{eb}^s	Successful EB transmission probability
p_{dio}^s	Successful DIO transmission probability
p_{dis}	Probability of DIS attack per slotframe
I_{min}	Minimum DIO generation interval
I_{max}	Maximum DIO generation interval
p_{dio}^i	DIO generation probability in state i
AJT	Average joining time
κ_i	Trickle resetting probability
C	number of transmitted DIO packet
T_{D_i}	current Trickle state of player J_i
k	Trickle redundancy constant
χ_i	Shared cell idle ratio

The generation and transmission of DIO packet in RPL-based LLNs are governed by the Trickle Algorithm [20]. According to the Trickle algorithm, upon receiving a multicast DIS request packet, a joined node resets its Trickle algorithm and starts generating a DIO packet starting from the minimum DIO generation interval, I_{min}. On the other hand, during normal network operation, the joined node doubles its DIO generation interval, I_c after the end of each interval (i.e., I_{c-1}) until the Trickle algorithm reaches its maximum DIO generation interval, I_{max}. The trickle algorithm doubles the DIO generation interval so that network bandwidth can be saved and nodes can save their energy by transmitting the minimum required DIO packet in the network. So, the generation (and so, transmission) of a DIO packet by the joined nodes is a sporadic process. Therefore, to find the DIO transmission probability in a shared cell, p_{dio}, we model Trickle algorithm's behavior in the presence of a DIS attack by an SMP. The proposed SMP is associated with discrete Markov states, which represent the different Trickle states, such as $I_{\text{min}}, I_1, \ldots I_c, \ldots I_{\text{max}}$ as shown in Fig. 1. The SMP is also associated with the time, $t_i = 2^{i-1} I_{\text{min}}$ spent on the Trickle state i, where $1 \leq i \leq T_n$ and T_n is total number of Trickle states, i.e., $I_{\text{max}} = 2^{T_n} I_{\text{min}}$. A joined node jumps from ith Trickle state to $(i+1)$th when there is no DIS attack in the network with probability $(1-p_{\text{dis}})$. Here, p_{dis} denotes the probability of DIS attack in a shared

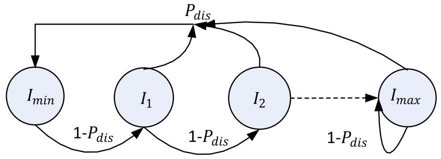


Fig. 1. Markov model of trickle algorithm with DIS attack.

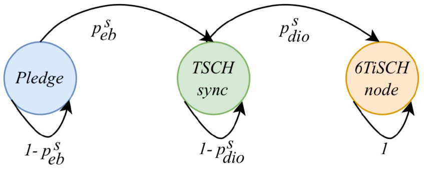


Fig. 2. Markov chain model of pledge joining process.

cell, which can be calculated as follows:

$$p_{\text{dis}} = \frac{L}{I_{\text{dis}}} \tag{1}$$

where L is the length of slotframe and I_{dis} is the DIS transmitting interval of a malicious node. As the joined nodes reset their Trickle algorithm immediately after receiving a multicast DIS request, p_{dis} becomes the DIS resetting probability of the joined nodes. Please note that for simplicity, we do not consider the other Trickle resetting conditions, such as DODAG version change, DODAG reset, and reception of inconsistent DIO packets. Now, the probability matrix of the Markov model can be written as follows:

$$P = \begin{bmatrix} p_{\text{dis}} & 1-p_{\text{dis}} & \dots & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{\text{dis}} & 0 & \dots & 1-p_{\text{dis}} & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ p_{\text{dis}} & 0 & \dots & 0 & 0 & 1-p_{\text{dis}} \end{bmatrix}.$$

In SMP, the probability, p_i of being in any state i is the product of the stationary probability of being in state i and the average amount of time the SMP spends in state i. Therefore, we can calculate p_i as follows:

$$p_i = \frac{p_{\text{dis}}(1-p_{\text{dis}})^{i-1} I_{\min} 2^{i-1}}{\sum_{k=1}^{T_n} p_{\text{dis}}(1-p_{\text{dis}})^{k-1} I_{\min} 2^{k-1}}. \tag{2}$$

So, the transmission probability of a DIO packet in a shared cell, p_{dio} can be calculated as follows:

$$p_{\text{dio}} = p_i \times p^i_{\text{dio}} \tag{3}$$

where p^i_{dio} is the DIO generation probability in state i, which is calculated as follows:

$$p^i_{\text{dio}} = \begin{cases} \frac{L}{2^{i-1} I_{\min}} & \text{if } \frac{L}{2^{i-1} I_{\min}} \geq 1 \\ 1 & \text{Otherwise.} \end{cases} \tag{4}$$

Now, to analyze the impact of the DIS attack on the 6TiSCH network, we consider the 6TiSCH network formation process with only one shared cell per slotframe as per the 6TiSCH-MC standard [4]. During 6TiSCH network formation, several control packets get exchanged among the pledges and already joined nodes in shared cells. However, efficient transmissions of both EB and DIO packet is essential for the faster formation of the 6TiSCH networks. Faster formation of 6TiSCH network helps in energy saving as nodes need to keep their radios active (which consumes maximum energy) before joining the network. Hence, we consider the 6TiSCH network formation scenario to show the impact of the DIS attack on 6TiSCH network.

During the formation of 6TiSCH networks, at the beginning, the pledges randomly scan all the channels (maximum 16 channels are used) one by one by keeping their radios active to receive the EB frame. This active scanning is required because the pledges do not know in which channel and when the already joined nodes transmit their control packets. However, after getting an EB frame, the pledges get synchronized with the underlying TSCH network, i.e., come to know about the channel and time when the following control packets would be transmitted. Once the pledges receive the DIO packet, they become 6TiSCH joined nodes, i.e., successfully joined the 6TiSCH network. Note that at the very beginning, there is only one joined node, known as join registrar/coordinator (JRC), who starts the formation process. Now, the states of a pledge during its network joining process, i.e., *pledge*, *TSCH synchronized node*, and *6TiSCH joined node* can be represented by the discrete-time Markov Chain model to find the average joining time (AJT) of the pledge. In brief, AJT of the pledge is nothing but the time taken by the pledge to reach the final *absorbing* state of the Markov model. The Markov model is shown in Fig. 2, where p^s_{eb} and p^s_{dio} denote the successful EB and DIO transmission probabilities of a joined node in a slotframe, respectively. The pledge jumps to the TSCH synchronized state after receiving an EB frame with the probability p^s_{eb} and jumps from TSCH synchronized state to the 6TiSCH joined node state with the probability p^s_{dio}. Therefore, AJT of a pledge, i.e., T_{avg} can be calculated as follows:

$$T_{\text{avg}} = \left(\frac{1}{p^s_{eb}} + \frac{1}{p^s_{\text{dio}}}\right) \times L. \tag{5}$$

Now, p^s_{eb} and p^s_{dio} can be calculated in presence of n joined nodes as follows:

$$p^s_{eb} = \frac{n}{N_c} p_{eb}(1-p_{eb})^{n-1}(1-p_{\text{dio}})^{n-1}(1-p_l) \tag{6}$$

$$p^s_{\text{dio}} = n p_{\text{dio}}(1-p_{eb})(1-p_{eb})^{n-1}(1-p_{\text{dio}})^{n-1}(1-p_l) \tag{7}$$

where p_{eb} denotes the probability of generating an EB frame per shared cell by the joined node, i.e., $p_{eb} = (L/I_{eb})$; I_{eb} is the EB generation interval of the joined node. On the other hand, the p_{dio} is calculated using (3). Before getting synchronized with the TSCH network, pledges randomly scan in all the available channels, so we divide the total probability by the total number of channels, i.e., N_c. Furthermore, the EB frame has a higher priority compared to DIO packet [4]. Hence, in (7), we use the term $(1 - p_{eb})$. The term p_l denotes the packet loss probability.

Now, to evaluate the DIS attack analytically, we consider the network settings as follows: $N_c = 16$, $L = 101$ timeslots, $t = 10$ ms, $I_{eb} = 4 * L$, $P_l = 0.2$, $I_{\min} = 8$ ms, $T_n = 16$. We compute AJT of a pledge using (5) by varying the number of nodes, n present in the network and DIS attacking rate,

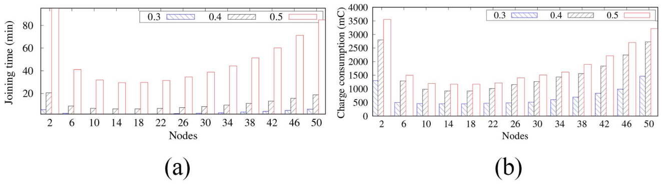


Fig. 3. Analytical results using different DIS attack probability per slotframe such that $p_{\text{dis}}=$ 0.3, 0.4, 0.5. (a) 6TiSCH joining time. (b) Charge consumption.

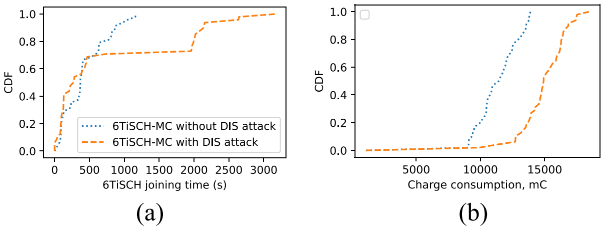


Fig. 4. Simulation results from Cooja Simulator. (a) 6TiSCH joining time. (b) Charge consumption.

p_{dis}. We also calculate the energy consumption (in terms of charge) of the pledge during its joining process by following the procedure mentioned in the work [16] considering OM-STM32 mote. In Fig. 3(a) and (b), we show the analytical results on 6TiSCH joining time and charge consumption of a pledge by varying the number of joined nodes and DIS attacking rate. Here, we can see that both the pledge's joining time and charge consumption increase with the increasing DIS attacking rate (i.e., highest with $p_{\text{dis}} = 0.5$). It happens because of the increasing congestion in a shared cell due to DIS attack, which creates burst transmission of DIO packets in the network. When the number of nodes in the network is less, the pledge needs to wait for more time to receive a valid EB frame as the transmitted EB frame collides with the frequently transmitted DIO packet. Hence, the joining time, as well as the charge consumption of the pledges, are high. On the other hand, when the number of nodes increases in the network, transmission of DIO packets due to DIS attack is also increased as more nodes reset their Trickle algorithm. This results in severe congestion in shared cell and increases the pledge's joining time and charge consumption drastically. These results turn out to be worst when the DIS attacking rate is high, i.e., $p_{\text{dis}} = 0.5$ in large networks.

We have also implemented the DIS attack on Contiki-NG and simulated the DIS attacking scenario on the Cooja simulator. For simulation, we consider a network with 49 nodes in a 7 × 7 Grid topology, in which node 1 is considered as JRC, nodes 5 and 23 are considered as malicious nodes, and the other 46 nodes are left as legitimate nodes. We run our simulation experiments for 1 h and collect the 6TiSCH joining time and charge consumption of the nodes considering the resource allocation proposed by 6TiSCH-MC standard [4]. The results are shown in Fig. 4(a) and (b). Our simulation results show that the DIS attack severely increases the joining time and charge consumption of the nodes. Some nodes, such as node 26, 40, and 49 fail to become joined nodes even after 1 h. Please note that we calculate the energy consumption by each node [i.e., Fig. 4(b)] during the entire simulation duration. On the other hand, we calculate its energy consumption in our analytical model till the pledge joining time. The Cooja simulator is more realistic compared to our analytical model. Our analytical model does not consider many network parameters and conditions, such as interference, multipath fading, and transmission range. Hence, the experimental results from both the analytical method and simulation experiments are different but show the same pattern or trend.

From the results of our theoretical and simulation experiments, we can conclude that the DIS attack severely affects the joining time and charge consumption of the nodes in 6TiSCH networks. Hence, in the following sections, we propose gaming and trust models to deal with the DIS attack in the 6TiSCH network.

V. Modeling of Noncooperative Gaming

In the 6TiSCH network, the joined nodes reset their Trickle algorithm to respond to the multicast DIS packet. It results in immediate and burst transmission of DIO packets in the network until all the joined nodes reach their maximum DIO generation Trickle interval. It is noteworthy that the joined nodes selfishly transmit their control packet, including the DIO packets. In brief, a joined node transmits its control packet without caring about the transmission of neighboring joined nodes and congestion in a shared cell. Hence, packet collision and congestion in shared cell increase drastically when multicast DIS packets are frequently transmitted in the 6TiSCH network or other RPL-based LLNs. So, both the congestion and collision become serious problems when the DIS packet is multicasted frequently, i.e., the DIS attack is imposed in 6TiSCH networks. Therefore, in this section, we propose a *noncooperative gaming* approach to find the optimal Trickle resetting probability to reduce the congestion in a shared cell, and so to reduce the impact of DIS attack on 6TiSCH networks. As only the joined nodes transmit DIO packets, hence, the proposed gaming model is used only by the joined nodes. After calculating the optimal Trickle resetting probability, the joined nodes find the trustworthiness of the received DIS packet using a *trust model* (discussed in the next section).

Let us consider a set of already joined nodes including the JRC, say $J = \{J_1, J_2, \ldots J_n\}$, where J_i denotes the ith joined node present in a 6TiSCH network. These nodes frequently transmit their DIO packets selfishly by resetting the Trickle algorithm upon receiving a DIS packet. Each selfish joined node and its Trickle resetting probability can be modeled as a noncooperative game $G = \{J, (\Psi_i)_{J_i \in J}, (\varphi_i)_{J_i \in J}\}$.

1) *Players:* The already joined nodes (i.e., $\forall J_i \in J$) are considered as *players* in our proposed game, G. It is because the joined nodes transmit DIO packets without any cooperation with the neighboring nodes.
2) *Strategy:* Ψ_i is the *strategy* of player J_i. Strategy of each player J_i is to decide the Trickle resetting probability, κ_i such that $0 \leq \kappa_i \leq 1$. Therefore, the strategy space of player J_i is $\Psi_i = [0, 1]$ and strategy space for the whole game is $\Psi = \prod_{i=1}^{n} \kappa_i$, $\forall J_i \in J$.

3) *Pay-Off Function:* φ_i denotes the *pay-off function* of each player J_i. Each player J_i tries to maximize its pay-off function by choosing the best value of κ_i over [0, 1].

The frequent and burst transmission of DIO packet due to DIS attack increases the congestion, so packet collision in a shared cell. Therefore, we include shared cell congestion status (SCC), number of transmitted DIO packets with respect to redundancy constant, and nodes' packet transmission priority as a *price functions* in our proposed pay-off function. We formulate the pay-off function of player J_i as follows:

$$\varphi_i(\kappa_i, \kappa_{-i}) = u_i(\kappa_i) - \frac{1}{1 - CC_i(\kappa_i, \kappa_{-i})} - C_i(\kappa_i) - P_i(\kappa_i) \quad (8)$$

where $\varphi_i(\kappa_i, \kappa_{-i})$ is the payoff function of player J_i; where κ_i is the Trickle resetting probability of player J_i and κ_{-i} denotes the Trickle resetting probabilities of other players except J_i. We describe different functions of our proposed payoff function, i.e., the other parameters of (8) as follows.

1) *Utility Function:* The term $u_i(\kappa_i)$ denotes the utility function of player J_i, where κ_i is Trickle resetting probability. The value of $u_i(\kappa_i)$ is considered as follows:

$$u_i(\kappa_i) = \log(\kappa_i + 1). \quad (9)$$

The utility function is directly proportional to the players' pay-off function, and we use logarithm to make the function strictly concave. We add one with the utility function so that the value of u_i does not become *infinity* when $\kappa_i = 0$.

2) *Congestion Cost Function:* The term $CC_i(\kappa_i, \kappa_{-i})$ denotes the congestion in a shared cell due to transmission of DIO packets by the joined nodes, i.e., players J_i, $\forall J_i \in J$. The value of $CC_i(\kappa_i, \kappa_{-i})$ is determined as follows:

$$\begin{aligned} CC_i(p_i, p_{-i}) &= \frac{\text{Busy shared cell}}{\text{Busy shared cell} + \text{Idle shared cell}} \\ &= \frac{\sum_{t=0}^{T} 1 - (1 - \kappa_i)^n}{\sum_{t=0}^{T} 1 - (1 - \kappa_i)^n + (1 - \kappa_i)^n} \\ &= 1 - (1 - \kappa_i)^n. \end{aligned} \quad (10)$$

Here, T denotes the time interval in terms of number of shared cells for measuring the SCC, and n denotes the number of neighbor nodes, including the node itself. Please note that we use the Trickle resetting probability for calculating SCC as the minimum DIO interval is less than the slotframe length, i.e., $I_{\min} < L$.

3) *Counter Cost Function:* In the Trickle algorithm, the term *redundancy constant* is used to restrict the number of DIO transmissions so that congestion and energy consumption can be reduced. However, according to RPL [5], this redundancy constant is not considered when the joined node receives DIS request. So, this is one reason for increasing DIO packet transmission when there is a DIS attack in the network. Therefore, we calculate the counter cost function with respect to default redundancy constant, k as follows:

$$C_i(\kappa_i) = \frac{\kappa_i C}{nk} \quad (11)$$

where C is a counter for the number of transmitted DIO packets in the current DODAG instance.

4) *Priority Cost Function:* It is defined by the term $P_i(\kappa_i)$ to distinguish between high and low priority joined nodes. The value of $P_i(\kappa_i)$ is calculated as follows:

$$P_i(\kappa i) = \frac{\kappa_i}{T_{D_i}} \quad (12)$$

where T_{D_i} denotes the current Trickle state of player J_i. We consider this cost function to give higher opportunities to the nodes in higher Trickle states than those in lower Trickle states. A node would be in a lower Trickle state when it has recently reset its Trickle algorithm.

After defining the different functions, such as u_i, CC_i, C_i, and P_i of the players J_i, $\forall J_i \in J$, the final pay-off function can be written using (8) as follows:

$$\varphi_i(\kappa_i, \kappa_{-i}) = \alpha_i(\log \kappa_i + 1) - \frac{\beta_i}{(1 - \kappa_i)^n} - \frac{\gamma_i \kappa_i C}{nk} - \frac{\delta_i \kappa_i}{T_{D_i}} \quad (13)$$

where, α_i, β_i, γ_i, and δ_i denote the preference parameters of the *utility function*, *congestion cost function*, *counter cost function*, and *priority cost function*, respectively.

A. Solution of the Game

A noncooperative game should have a unique NE point to have a unique solution. We prove it for our proposed game $G = \{J, (\Psi_i)_{J_i \in J}, (\varphi_i)_{J_i \in J}\}$, through two Lemmas, i.e., Lemmas 1 and 2 which are presented as follows.

Lemma 1: The proposed pay-off function $\varphi_i(\kappa_i, \kappa_{-i})$; $\forall J_i \in J$ holds the concave property, and the game $G = \{J, (\Psi_i)_{J_i \in J}, (\varphi_i)_{J_i \in J}\}$ has at least one NE point.

Proof: The proof of this lemma is similar to the proof of [33, Lemma 1] and is also available in the Appendix. ■

Lemma 2: The proposed game G has a unique solution as it has a unique NE point.

Proof: The proof of this lemma is similar to the proof of [33, Lemma 2] and is also available in the Appendix. ■

Lemmas 1 and 2 have proved that our proposed game G holds a unique NE point and a unique solution. Therefore, in order to solve G, we construct the following *constrained nonlinear optimization* problem:

$$\begin{aligned} &\underset{p_i \in S_i}{\text{maximize}} \quad \varphi_i(\kappa_i, \kappa_{-i}) \\ &\text{subject to} \\ &0 \leq \kappa_i; \forall J_i \in J \\ &0 \leq \kappa_i^{\max} - \kappa_i; \forall J_i \in J. \end{aligned} \quad (14)$$

Now, to solve this constrained nonlinear optimization problem, we can write the *Lagrange function* $L_i(\kappa_i, \kappa_{-i}, \sigma_i)$ for player J_i, $\forall J_i \in J$, as follows:

$$L_i(\kappa_i, \kappa_{-i}, \sigma_i) = \varphi_i(\kappa_i, \kappa_{-i}) + \nu_i \kappa_i + \sigma_i\left(\kappa_i^{\max} - \kappa_i\right) \quad (15)$$

where, ν_i and σ_i are two Lagrange multipliers constant. Now, we can write the KKT conditions as follows:

$$\begin{aligned} &\nu_i, \sigma_i \geq 0 \\ &\kappa_i \geq 0 \\ &\kappa_i^{\max} - \kappa_i \geq 0 \\ &\nabla_{\kappa_i} \varphi_i(\kappa_i, \kappa_{-i}) + \nu_i \nabla_{\kappa_i} (\kappa_i) + \sigma_i \nabla_{\kappa_i} \left(\kappa_i^{\max} - \kappa_i\right) = 0 \end{aligned}$$

$$\nu_i \kappa_i = 0$$
$$\sigma_i\left(\kappa_i^{\max} - \kappa_i\right) = 0.$$

Equation (15) has three unknown variables, i.e., κ_i, ν_i, and σ_i. We use KKT conditions to calculate the optimal Trickle resetting probability (κ_i^*) of player J_i as follows:

$$\kappa_i^* = \begin{cases} \rho_i^{\min}; & \text{if condition 1} \\ \rho_i^{\max}; & \text{if condition 2} \\ \frac{\alpha_i}{\frac{n\beta_i}{\chi_i} + \frac{\gamma_i}{T_{D_i}} + \frac{\delta_i \times C}{n \times k}} - 1; & \text{otherwise} \end{cases} \tag{16}$$

where condition 1 is

$$\frac{n\beta_i}{\alpha_i - \frac{\gamma_i}{T_{D_i}} - \frac{\delta_i \times C}{n \times k}} \geq \chi_i. \tag{17}$$

The term χ_i denotes the *shared cell idle ratio*, which is nothing but $\chi_i = (1 - p_i)^{n+1}$. We derived the condition 1 considering $\kappa_i = 0$ and $\sigma_i = 0$, which implies

$$\Rightarrow \frac{\alpha_i}{\kappa_i + 1} - \frac{n\beta_i}{(1-\kappa_i)^{n+1}} - \frac{\gamma_i}{T_{D_i}} - \frac{\delta_i C}{nk} + \nu_i = 0$$
$$\Rightarrow \nu_i = -\frac{\alpha_i}{\kappa_i + 1} + \frac{n\beta_i}{(1-\kappa_i)^{n+1}} + \frac{\gamma_i}{T_{D_i}} + \frac{\delta_i C}{nk}. \tag{18}$$

The solution $\kappa_i = 0$ is feasible, if $\nu_i > 0$ holds, therefore

$$-\frac{\alpha_i}{\kappa_i + 1} + \frac{n\beta_i}{(1-\kappa_i)^{n+1}} + \frac{\gamma_i}{T_{D_i}} + \frac{\delta_i C}{nk} \geq 0$$
$$\frac{n\beta_i}{\alpha_i - \frac{\gamma_i}{T_{D_i}} - \frac{\delta_i C}{nk}} \geq \chi_i. \tag{19}$$

Similarly, condition 2 is obtained by considering $\kappa_i = \kappa_i{}^{\max}$, $\nu_i = 0$, and, here, also $\kappa_i = \kappa_i{}^{\max}$ is feasible only when the condition, $\sigma_i > 0$ holds. Therefore, the final obtained condition 2 is as follows:

$$\frac{n\beta_i}{\frac{\alpha_i}{p_i{}^{\max}} - \frac{\gamma_i}{T_{D_i}} - \frac{\delta_i \times C}{n \times k}} \leq \chi_i. \tag{20}$$

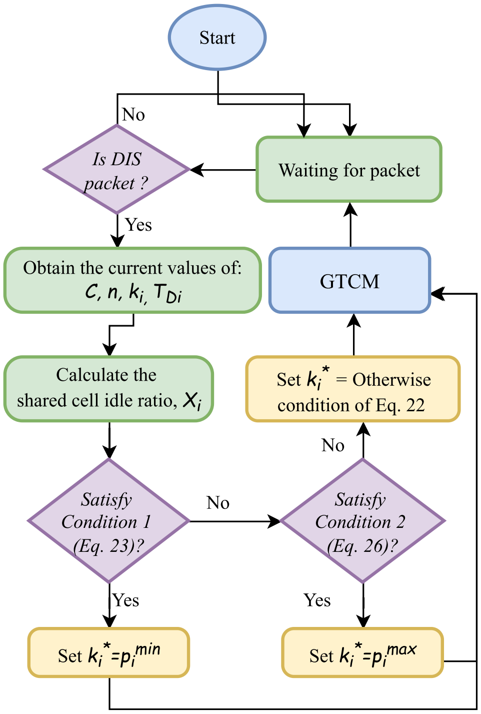
Finally, the *otherwise* condition holds all the conditions that do not fall under conditions 1 and 2. The *otherwise* condition is obtained considering $\nu_i = 0$, $\sigma_i = 0$, and $0 < \kappa_i < \kappa_{\max}$. In Fig. 5, we provide the flow chart of the proposed gaming model.

VI. Trust Model

This section describes our proposed trust model to distinguish maliciously transmitted DIS packets from normal DIS packets in a given 6TiSCH network. For this, we calculate a term called trust factor (TF) for each transmitted DIS packet in the network by considering the following three aspects.

1) *DIS Transmitting Rate:* If a node frequently transmits DIS packet compared to the standardized DIS rate mentioned in [20], then we assign binary value to the variable $X = 1$, otherwise, $X = 0$. Here, the true value of binary variable X denotes that the node maliciously transmitted DIS request, i.e., at a higher rate. The standardized DIS rate is $1/60$ sec [5]. So, we consider this value as a threshold value for comparison.

2) *Is It a Joined Node?* The joined node checks whether or not the DIS transmitting node has transmitted the data packet before. If the node has transmitted data packets before, then we assign $Y = 0$, otherwise $Y = 1$. We consider this condition because a malicious node always tries to perform its activity for a longer duration. Hence, it would not waste energy by transmitting data packets.



Start
No
Is DIS packet ?
Waiting for packet
Yes
Obtain the current values of: C, n, k_i, T_Di
GTCM
Calculate the shared cell idle ratio, X_i
Set k_i* = Otherwise condition of Eq. 22
No
Satisfy Condition 1 (Eq. 23)?
No
Satisfy Condition 2 (Eq. 26)?
Yes
Yes
Set k_i*=p_i^min
Set k_i*=p_i^max

Fig. 5. Working of the proposed gaming model.

3) *Information From RSSI:* A malicious node can perform DIS attack by changing its `EUI64` addresses which is known as Sybil attack [30]. Hence, we assign binary value to the variable $Z = 1$ when a node receives multiple DIS requests with the same RSSI value but different `EUI64` addresses. Otherwise, we assign $Z = 0$. We can assume that the nodes in any IoT network are deployed at least at a minimum distance which does not incur the same RSSI value to reduce the deployment cost.

After assigning binary values to the above-mentioned three binary variables, we calculate the final TF as follows:

$$\text{TF} = 1 - \left(\omega_x X + \omega_y Y + \omega_z Z\right) \tag{21}$$

where ω_x, ω_y, and ω_z are the weight parameters associated with the above-mentioned three binary variables such that $\omega_x + \omega_y + \omega_z = 1$ and $\omega_x = 0.5$, $\omega_y = 0.25$, and $\omega_z = 0.25$. We assign the highest weight to ω_x because if a node transmits a DIS request with more than the standardized rate, then the node is malicious. On the other hand, we assign $\omega_y = 0.25$ and $\omega_z = 0.25$ because at the beginning, a node would not transmit any data packet, and there may be a case that a joined node may receive multiple DIS requests from multiple nodes, which are located very closely. At the beginning, the values of X, Y, and Z for all the nodes are considered as 0, i.e., we consider all the nodes as legitimate nodes. These values get changed depending on the activities of the nodes after joining the network, as nodes can transmit their DIS packets only after joining the network. Hence, the final TF will be in TF = [0, 1]

Algorithm 1 GTCM

```
1: **INPUT**: $\kappa_i$, $\kappa_{-i}$, $C$, $k$, $T_{D_i}$, $X$, $Y$, $Z$
2: **OUTPUT**: $\kappa_i^*$
3: **if** Received multicast DIS packet **then**
4:     Calculate *shared cell idle probability* i.e., $\chi_i$
5:     Calculate $\kappa_i^*$ using the Equation (16)
6:     Calculate *Trust factor*, *TF* using the Equation (21)
7:     **if** *Trust factor*, *TF* $> 0.5$ **then**
8:         Reset the Trickle algorithm with probability $\kappa_i^*$
9:     **else**
10:        Ignore the DIS request
11:        (**mark the sender as malicious node*)
12:    **end if**
13: **end if**
```

for every possible value of above mentioned binary variables. So, after calculating the value of TF, if a joined node finds that TF ≤ 0.5, then it considers that a malicious node transmits the received DIS request, otherwise, it is considered as a normal DIS request, i.e., when TF > 0.5.

VII. Gaming and Trust-Based Countermeasure

In this section, we describe our proposed scheme GTCM. GTCM uses both the proposed *noncooperative gaming model* and *trust model* to reduce the effect of DIS attack on 6TiSCH networks. Using the gaming model's solution, GTCM decides the optimal Trickle resetting probability, and using the trust model, GTCM finds the presence of maliciously transmitted DIS packets in the 6TiSCH networks. Please note that only joined nodes run the GTCM scheme. After receiving a multicast DIS request, a joined node calculates the SCC from when it received its last DIS request. Note that SCC is measured without any signaling overhead in the network. It is because all the nodes need to keep their radios active in the shared cell for transmitting or receiving control packets in 6TiSCH network. SCC is measured by dividing the total number of shared cells in which the joined node has either received or transmitted control packet by the total number of shared cells in the measuring interval. Now, by subtracting the value of SCC from 1, we can calculate the value of χ_i, which is used (16). Finally, using (16), we can calculate the optimal Trickle resetting probability, κ_i^*.

After calculating the value of κ_i^*, we use our trust model to find maliciously transmitted DIS packets in the network. For this, we use (21). If the proposed trust model classifies the current DIS packet as malicious, the joined node ignores the received DIS packet. Otherwise, the joined node resets its Trickle algorithm with the probability κ_i^*, which is calculated using the gaming model. Please note that after classifying a DIS packet as a malicious packet, the joined nodes can take necessary action against the sender, such as blocking it permanently, report to JRC or the network administrator. However, these techniques are not included in this work. Algorithm 1 describes the steps of the proposed GTCM, where step 9 is not considered in this work.

VIII. Performance Evaluation

We implement our proposed scheme GTCM on Contiki-NG and create binary files (Contiki OS) for 32-bit ARM Cortex M3 microcontroller-based IoT node. We use the open-source FIT IoT-LAB to perform our testbed experiments, where we select 62 nodes from the *Strasbourg* location. The nodes are deployed in a 3-D space as shown in Fig. 6, and all the nodes used all the 16 communication channels with -17-dBm packet transmission (Tx) power. Out of the selected 62 nodes, we consider node 38 as JRC, node 36, and 24 as malicious nodes who maliciously transmit DIS packets after every second. The network formation process is started by the JRC, i.e., node 38 and the other nodes join the network once they receive both the EB and DIO control packets. The other parameters used in our testbed experiment are mentioned in Table IV.

TABLE IV
Experimental Settings

Symbol	Meaning
n	Number of nodes
N_c	Total number of channels used
L	Length of slorframe
I_{eb}	EB generation interval
p_{eb}	EB generation probability in a shared cell
P_{dio}	DIO generation probability in a shared cell
p_l	Packet loss probability
p_{eb}^s	Successful EB transmission probability
p_{dio}^s	Successful DIO transmission probability
p_{dis}	Probability of DIS attack per slotframe
I_{min}	Minimum DIO generation interval
I_{max}	Maximum DIO generation interval
p_{dio}^i	DIO generation probability in state i
AJT	Average joining time
κ_i	Trickle resetting probability
C	number of transmitted DIO packet
T_{D_i}	current Trickle state of player J_i
k	Trickle redundancy constant
χ_i	Shared cell idle ratio

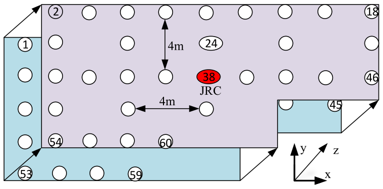
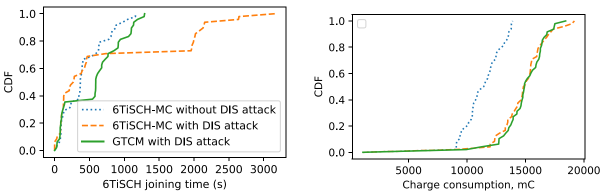


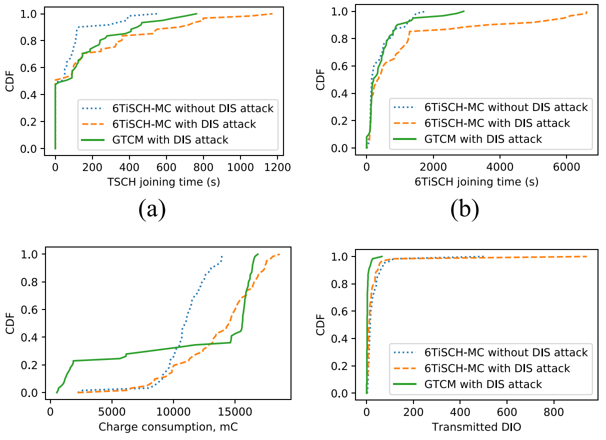
Fig. 6. Strasbourg FIT IoT-LAB testbed with 62 M3 nodes.

We evaluate the impact of DIS attack on the considered 6TiSCH network (shown in Fig. 6) in terms of the matrices mentioned in the previous Section IV, i.e., 6TiSCH joining time and energy consumption. However, apart from these two metrics, we also consider other metrics, such as TSCH synchronization time, number of transmitted DIO, and DIS packets during our testbed experiments. TSCH synchronization time of a pledge is the time the pledge takes to receive its EB frame. The pledge consumes maximum energy before getting synchronized with the TSCH network as it needs to keep its radio active to get an EB frame. On the other hand, the time is taken by a TSCH synchronized node to receive a valid DIO packet (i.e., to become a 6TiSCH joined node) is considered as 6TiSCH joining time. We further measure the amount of charge consumed by each node during our testbed experiments as the DIS attack increases the energy consumption of the nodes by increasing the transmission of DIO packets in the 6TiSCH



(a) (b)

Fig. 7. Simulation results from Cooja Simulator. (a) 6TiSCH joining time. (b) Charge consumption.



(a) (b)

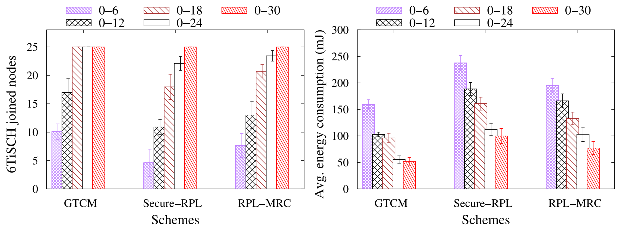
(c) (d)

Fig. 8. Testbed results from FIT IoT-LAB. (a) TSCH sync time. (b) 6TiSCH joining time. (c) Charge consumption. (d) Transmitted DIO.

networks. For measuring the charge consumption by the nodes, we use the *Energest module* available on Contiki-NG.

We also provide the simulation-based experimental results of the proposed scheme using the previous 7×7 topology along with testbed results. The simulation results are shown in Fig. 7 and the testbed experimental results are shown in Figs. 8 and 9. From the simulation results, it can be observed that nodes take less time to join the network [Fig. 9(a)] and consume less charge [Fig. 9(b)] in presence of DIS attack using the proposed scheme GTCM compared to the default (i.e., GTCM is not used) network scenario. The testbed results on TSCH synchronization time and 6TiSCH joining time shown in Fig. 8(a) and (b), respectively, show that GTCM reduces both the joining times of the nodes even when there is a DIS attack in the network. GTCM shows better results because it responds to the DIS packets only when the legitimate nodes have transmitted those DIS packets. Furthermore, to balance the shared cell congestion (SCC) and maintain fair DIO transmission opportunity among nodes, GTCM resets the nodes' Trickle algorithm with optimal probability depending on the current SCC and nodes' Trickle state. Fig. 8(d) shows that GTCM transmits less number of DIO packets compared to the other two network conditions. In brief, GTCM can balance the number of transmitted DIO packets in the network, congestion in a shared cell, and fair transmission of DIO packet. So, because of the balanced congestion in a shared cell, GTCM can reduce both the nodes' TSCH and 6TiSCH joining times.

In Fig. 8(c), we show the energy consumption of the nodes during experiments. Here, we can see that GTCM is able to



(a) (b)

Fig. 9. Comparison with existing DIS attack detection schemes for from 5 × 5 Grid topology. The bar shows the number of joined nodes and average energy consumption by the nodes for various time intervals (e.g., 0–6, 0–12, etc., in minutes). (a) 6TiSCH joining time. (b) Energy consumption.

reduce the energy consumption of the nodes to some extent, but nodes also consume more energy compared to when there is no DIS attack in the network using GTCM. It is because, even though GTCM responds optimally to DIS packets, the nodes receive the DIS packets transmitted by the malicious nodes. As nodes consume energy for receiving packets also, GTCM fails to reduce the energy consumption of the nodes. However, it saves some energy by transmitting a minimum number of DIS packets in the network compared to the network with DIS attacks.

In Fig. 9, we compare our proposed scheme GTCM with the recent existing schemes Secure-RPL [13] and RPL-MRC [32] on DIS attack detection and mitigation. Please note that we obtain these results using a 5 × 5 Grid topology in *Strasbourg*. We run each experiment several times for 30 min and provide the results with 95% confidence interval. The results show that GTCM performs better than both existing schemes. It is because, Secure-RPL [13] and RPL-MRC [32] did not consider the Sybil DIS attack, i.e., when the malicious nodes transmit their DIS packets with fictitious `EUI64` id. Even though, RPL-MRC [32] can reduce congestion in a shared cell by dynamically varying the DIO transmitting intervals compared to Secure-RPL [13], it fails to reduce congestion as much as GTCM can do. We also note that GTCM also considers congestion in a shared cell while calculating the DIS response probability apart from other parameters. Hence, GTCM performs better than both existing schemes. On the other hand, Secure-RPL [13] shows the worst performance as it neither considers the Sybil DIS attack nor varies the DIO transmission interval.

IX. Conclusion

In this work, we proposed a noncooperative gaming and trust model-based countermeasure for DIS attacks considering 6TiSCH IoT networks. We formulated a noncooperative gaming model with the joined nodes as players. In the pay-off function, we considered the Trickle resetting probability as a *utility function*. As a shared cell gets congested due to the transmission of more DIO packets in the network, we considered the SCC as one of the parameters for *price function*. Furthermore, to provide equal opportunity, we used two more parameters (i.e., number of transmitted DIO packets in an interval (counter) and Trickle state of a joined node) for the price function. Apart from this gaming model, we also proposed a trust model to find malicious transmissions of DIS packets in 6TiSCH networks. Finally, we combined the solution of our gaming model and trust model and proposed the GTCM to reduce the effect of DIS attacks on 6TiSCH networks. The proposed scheme GTCM was implemented on Contiki-NG and evaluated using FIT IoT-LAB testbed. The testbed experiment results show that GTCM reduces the effect of DIS attacks in terms of pledges' joining time and energy consumption significantly.

References

[1] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2015 (Revision IEEE Std 802.15.4-2011), Apr. 2016.

[2] X. Vilajosana et al., "IETF 6TiSCH: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 1–21, 1st Quart., 2020.

[3] T. Watteyne et al., "Industrial wireless IP-based cyber–physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1025–1038, May 2016.

[4] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH mode of IEEE 802.15.4e (6TiSCH) configuration," Internet Eng. Task Force, RFC 8180, May 2017.

[5] T. Winter et al., "RPL: IPv6 routing protocol for low-power and lossy networks," Internet Eng. Task Force, RFC 6550, Mar. 2012.

[6] A. Raoof, A. Matrawy, and C.-H. Lung, "Routing attacks and mitigation methods for RPL-based Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1582–1606, 2nd Quart., 2019.

[7] P. Perazzo, C. Vallati, G. Anastasi, and G. Dini, "DIO suppression attack against routing in the Internet of Things," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2524–2527, Nov. 2017.

[8] A. Jain and S. Jain, "A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT," in *Emerging Technologies in Data Mining and Information Security*. Singapore: Springer, 2019, pp. 611–620.

[9] F. Medjek, D. Tandjaoui, I. Romdhani, and N. Djedjig, "A trust-based intrusion detection system for mobile RPL based networks," in *Proc. IEEE Int. Conf. Internet Things*, 2017, pp. 735–742.

[10] A. Kalita, A. Brighente, M. Khatua, and M. Conti, "Effect of DIS attack on 6TiSCH network formation," *IEEE Commun. Lett.*, vol. 26, no. 5, pp. 1190–1193, May 2022.

[11] C. Pu, "Spam DIS attack against routing protocol in the Internet of Things," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2019, pp. 73–77.

[12] B. Farzaneh, M. A. Montazeri, and S. Jamali, "An anomaly-based IDS for detecting attacks in RPL-based Internet of Things," in *Proc. Int. Conf. Web Res. (ICWR)*, 2019, pp. 61–66.

[13] A. Verma and V. Ranga, "Mitigation of DIS flooding attacks in RPL-based 6LoWPAN networks," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, Feb. 2020, Art. no. e3802.

[14] *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 15.1a: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Wireless Personal Area Networks (WPAN)*, IEEE Standard 802.15.1-2005 (Revision IEEE Std 802.15.1-2002), 2005.

[15] C. Vallati, S. Brienza, G. Anastasi, and S. K. Dass, "Improving network formation in 6TiSCH networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 1, pp. 98–110, Jan. 2019.

[16] A. Kalita and M. Khatua, "Channel condition based dynamic beacon interval for faster formation of 6TiSCH network," *IEEE Trans. Mobile Comput.*, vol. 20, no. 7, pp. 2326–2337, Jul. 2021.

[17] A. Kalita and M. Khatua, "Opportunistic transmission of control packets for faster formation of 6TiSCH network," *ACM Trans. Internet Things*, vol. 2, no. 1, p. 5, Jan. 2021.

[18] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, 2004, pp. 455–462.

[19] C. Adjih et al., "FIT IoT-LAB: A large scale open experimental IoT testbed," in *Proc. IEEE 2nd World Forum Internet Things*, Dec. 2015, pp. 459–464.

[20] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm," Internet Eng. Task Force, RFC 6206, Mar. 2011.

[21] P. Perazzo, C. Vallati, A. Arena, G. Anastasi, and G. Dini, "An implementation and evaluation of the security features of RPL," in *Ad-Hoc, Mobile, Wireless Networks*. Cham, Switzerland: Springer, 2017, pp. 63–76.

[22] C. Gündoğan, D. Barthel, and E. Baccelli, "DIS modifications," Internet-Draft draft-ietf-roll-dis-modifications-01, Internet Eng. Task Force, Fremont, CA, USA, Nov. 2019.

[23] F. Meneghello, M. Calore, D. Zucchetto, M. Polese, and A. Zanella, "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8182–8201, Oct. 2019.

[24] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of IoT new features on security and privacy: New threats, existing solutions, and challenges yet to be solved," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[25] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the Internet-of-Things networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021.

[26] S. I. Popoola, R. Ande, B. Adebisi, G. Gui, M. Hammoudeh, and O. Jogunola, "Federated deep learning for zero-day botnet attack detection in IoT-edge devices," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3930–3944, Mar. 2022.

[27] S. Sharma and V. K. Verma, "Security explorations for routing attacks in low power networks on Internet of Things," *J. Supercomputing*, vol. 77, no. 5, pp. 4778–4812, 2021.

[28] P. P. Ioulianou and V. G. Vassilakis, "Denial-of-service attacks and countermeasures in the RPL-based Internet of Things," in *Computer Security*. Cham, Switzerland: Springer, 2020, pp. 374–390.

[29] C. Pu and K.-K. R. Choo, "Lightweight Sybil attack detection in IoT based on bloom filter and physical unclonable function," *Comput. Security*, vol. 113, Feb. 2022, Art. no. 102541.

[30] C. Pu, "Sybil attack in RPL-based Internet of Things: Analysis and defenses," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4937–4949, Jun. 2020.

[31] A. Alsirhani et al., "Securing low-power blockchain-enabled IoT devices against energy depletion attack," *ACM Trans. Internet Technol.*, to be published.

[32] F. Medjek, D. Tandjaoui, N. Djedjig, and I. Romdhani, "Multicast DIS attack mitigation in RPL-based IoT-LLNs," *J. Inf. Security Appl.*, vol. 61, Sep. 2021, Art. no. 102939.

[33] A. Kalita and M. Khatua, "A noncooperative gaming approach for control packet transmission in 6TiSCH network," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3954–3961, Mar. 2022.

Alakesh Kalita (Member, IEEE) received the B.Tech. degree from Assam Don Bosco University, Guwahati, India, in 2012, the M.Tech. degree from Assam University, Silchar, India, in 2016, and the Ph.D. degree in computer science and engineering from the Indian Institute of Technology Guwahati, Guwahati, India, in 2022.

He is currently a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include Internet of Things and edge/cloud computing.

Mohan Gurusamy (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Indian Institute of Technology Madras, Chennai, India, in 2000.

In June 2000, he joined the National University of Singapore, Singapore, where he is currently an Associate Professor with the Department of Electrical and Computer Engineering. He has about 220 publications to his credit, including two books and three book chapters in the area of optical networks. His research experience and interests are in the areas of Internet of Things, 5G networks, software-defined networks, network function virtualization, cloud computing, data center networks, and optical networks.

Dr. Gurusamy served as a TPC co-chair for several conferences and served as an Editor for IEEE TRANSACTIONS ON CLOUD COMPUTING and is serving on the editorial board for *Computer Networks* (Elsevier) and *Photonic Network Communications* (Springer).

Manas Khatua (Member, IEEE) received the Ph.D. degree from Indian Institute of Technology (IIT) Kharagpur, Kharagpur, India, in 2015.

He has been an Assistant Professor with the Department of Computer Science and Engineering, IIT Guwahati, Guwahati, India, since May 2018. Prior to that, he was an Assistant Professor with IIT Jodhpur, Jodhpur, India, from 2016 to 2018, and was a Postdoctoral Research Fellow with SUTD, Singapore, from 2015 to 2016. He was associated with Tata Consultancy Services, Kolkata, India, from 2008 to 2010. His research interests include performance evaluation of communication protocols, Internet of Things, wireless LANs, sensor networks, and network security.