Journal of Network and Computer Applications 209 (2023) 103516

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Secured location-aware mobility-enabled RPL

Erfan Arvan, Mahshad Koohi Habibi Dehkordi, Saeed Jalili *

Computer Engineering Department, Tarbiat Modares University, Tehran, Iran

ARTICLE INFO

Index Terms:
Internet of things (IoT)
RPL
Mobility management
Security
Intrusion detection system

ABSTRACT

Although there have been many studies on the Internet of Things (IoT), there are still major challenges for IoT to become ubiquitous. So far, the mobility management challenge has not been addressed well. Routing Protocol for Low-Power and Lossy Networks (RPL), which is known as the de-facto for routing in IoT, does not support mobile nodes. Some studies have tried to address this challenge, but they either caused a very high Packet Loss Rate (PLR) or produced lots of control packets. Also, they have not considered the security aspects of their work which is crucial for real-world applications. In this study, a novel extension for the RPL called Secured Location-Aware Mobility-enabled RPL (SLM-RPL) is proposed to facilitate the mobility management of RPL while considering security precautions. From the mobility management point of view, according to extensive evaluations, SLM-RPL greatly reduces the hand-off delay and PLR compared to other mobility management schemes, even in big, dense, or highly dynamic networks. Therefore, SLM-RPL is shown to be the best option to be used in IoT applications, especially loss-sensitive ones. Also, SLM-RPL produces small numbers of control packets and has a low memory overhead. Moreover, from the security point of view, a probability-based method has been proposed and embedded in SLM-RPL, which is shown to be able to reduce the negative impacts of DODAG Information Solicitation (DIS) attacks by more than 99%. Also, a performable attack on SLM-RPL called False-Location-Injection (FLI) attack has been introduced, and a lightweight hybrid-structured Intrusion Detection System (IDS) has been provided to counter this attack as well as Sybil, Rank, Sinkhole, and impersonation attacks. The proposed IDS uses a voting-based approach which, when the ψ parameter is adjusted, can mitigate the impact of False-Reporting and collusion attacks. According to the evaluations, the proposed IDS can counter the mentioned attacks in the presence of Collusion attackers in different scenarios with Accuracy $\geq$0.99.

1. Introduction

The Internet of things (IoT) is an ecosystem of technologies that enables many useful applications such as healthcare, smart city, smart home, industrial, agriculture, transport, logistics, and healthcare. Many of these applications include some mobile things (objects or IoT devices)

In order to prevent reinventing the wheel, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) (Montenegro et al., 2007) has been proposed as a bridge between the existing IP world means the Internet Protocol version 6 (IPv6) (Committee, 2003) and IoT small devices (IEEE 802.15.4 standard (Committee, 2003)); therefore, it can be said, with a little bit of neglect, we can develop IoT using the current global Internet platform. However, as the routing procedures of IPv6 are too heavy to run on small resource-constrained IoT devices (i.e., low-cost devices that have limited processing, storage, and networking capabilities and often run on batteries) (Yugha and Chithra, 2020), Routing Protocol for Low-Power and Lossy Networks (RPL) has been proposed by the Routing Over Low-power and Lossy Networks (ROLL) working group of the Internet Engineering Task Force (IETF) in RFC-6550 (Winter et al., 2012). Afterward, RPL has been known as the de-facto for routing in IoT.

RPL is a distance-vector routing protocol that operates on top of the 6LoWPAN and the IEEE 802.15.4 standard. This protocol is a light-weight solution for routing in Low-power and Lossy Networks (LLN) (Winter et al., 2012). RPL network is formed in a tree-like structure called Destination-Oriented Directed Acyclic Graph (DODAG), including a root node (i.e., a powerful node that plays the role of sink or gateway of the network) responsible for building and maintaining the network topology (Gaddour and Koubâa, 2012).

Although RPL is the most prominent routing protocol for IoT, it does not support the mobility of the nodes. So far, several extensions have been developed to support mobility in RPL, Such as ME-RPL (El Korbi

Abbreviations: RPL, Routing Protocol for Low-Power and Lossy Networks; IDS, Intrusion Detection System; IoT, Internet of Things.

* Corresponding author.
E-mail addresses: e.arvan@yahoo.com (E. Arvan), mahshadkoohi@gmail.com (M. Koohi Habibi Dehkordi), sjalili@modares.ac.ir (S. Jalili).

https://doi.org/10.1016/j.jnca.2022.103516
Received 11 December 2021; Received in revised form 22 July 2022; Accepted 11 September 2022
Available online 14 October 2022

et al., 2012), MMRPL (Cobarzan et al., 2014), CO-RPL (Gaddour et al., 2014), MRPL-V (Lee et al., 2012), DMR (Hong and Choi, 2011), and mRPL (Fotouhi et al., 2015), mRPL+ (Fotouhi et al., 2017) and Manikannan and Nagarajan (2020). Moreover, during examining mobility-enabled extensions of RPL, it is concluded that either the Packet Loss Rate (PLR) in these protocols is much higher than what is needed in the real world, or they cause too much control overhead. Therefore, the Secured Location-Aware Mobility-enabled RPL (SLM-RPL) is proposed to overcome these problems. SLM-RPL is a location-aware protocol which means the mobile nodes use the location of their neighbors to calculate the distance to them and manage the mobility more accurately.

The main focus of this paper is to deal with the mobility management challenge of RPL; however, we believe that the security aspects of a mobility extension should be considered not only because some attacks can decrease the performance of the mobility extension to a great deal, but also because some vulnerabilities may stem from the proposed mobility extension architectures (e.g., False Location Injection attack which will be explained in the Security Considerations section). Security is another significant challenge for the IoT to become ubiquitous (Almusaylim et al., 2020; Mahbub, 2020). Obviously, there are no benefits when we are not secured. Unfortunately, although there have been scattered studies on RPL security (Almusaylim et al., 2020; Agiollo et al., 2021; Airehrour et al., 2019; Baghani et al., 2020; Ghaleb et al., 2018; Guo, 2021; Kamble et al., 2017; Le et al., 2016; Murali and Jamalipour, 2019; Osman et al., 2021; Perazzo et al., 2017; Pu, 2018, 2020; Sharma et al., 2021; Simoglou et al., 2021; Thulasiraman and Wang, 2019; Tsao et al., 2015; Verma and Ranga, 2019, 2020a, 2020b), none have provided a comprehensive approach to enable security on mobility extensions or in dynamic (i.e., mobility-contained) networks. Consequently, unlike other mobility extensions, one of the most important contributions of SLM-RPL is to consider the security aspects. In this regard, first, a probability-based method is proposed to mitigate the negative impacts of DODAG Information Solicitation (DIS) attacks (Verma and Ranga, 2020b), which are one of the crucial vulnerabilities of RPL mobility extensions. Then a new attack called False-Location-Injection (FLI) is introduced on SLM-RPL. Afterward, a lightweight hybrid-structured Intrusion Detection System (IDS) is proposed to counter FLI, Sybil, Rank, Sinkhole, and Impersonation attacks in dynamic networks. It can also dramatically mitigate the possibility of successfully performing False-Reporting and Collusion attacks.

Accordingly, the most important contributions of this study can be summarized as follows:

1) Providing a new mobility extension for RPL called SLMRPL, which has the lowest hand-off delay and Packet Loss Rate (PLR) among other mobility extensions
2) Effectively managing the mobility without causing high power consumption and memory overhead
3) Providing a robust probability-based approach to mitigate the negative impacts of DIS attacks
4) Considering the security aspects and providing a lightweight IDS to detect FLI, Sybil, rank/sinkhole, and impersonation attacks on dynamic networks.

The rest of this paper is organized as follows. Section II overviews the RPL. Then, in section III, related work is presented. Section IV describes SLM-RPL, and section V explains the security considerations. Then, in section VI, the evaluation of SLM-RPL in mobility management and security is provided. Finally, Section VIII presents the conclusion.

2. RPL

RPL (Winter et al., 2012) is specially designed to run on resource-constrained nodes to manage the interoperations between IoT network nodes. An RPL network is formed as a DODAG with a root responsible for managing the network. In a steady-state, each node in the network has a preferred parent, and potential parents set. The preferred parent and potential parents set are determined by using an objective function (OF). Each node in the network has a rank that specifies the node's position relative to the other nodes in the root path. The role of the objective function is to rank RPL nodes by using one or more criteria, then choosing the best one as the preferred parent. RPL involves four control messages named DODAG Information Object (DIO), DODAG Information Solicitation (DIS), Destination Advertisement Object (DAO), and Destination Advertisement Object Acknowledgement (DAO-ACK) (Gaddour and Koubâa, 2012). Also, there are three types of nodes in RPL:

1) **Root**, which is responsible for constructing the DODAG
2) **Routers**, which can be the preferred parent of other nodes
3) **Leaf nodes** which cannot be preferred parents

DODAG construction uses two primary operations: (1) broadcasting DIO messages to form upward routes and (2) unicasting DAO messages up to the root to form downward routes.

The root node starts constructing the DODAG by broadcasting a DIO message. By receiving this message, all of the root neighbors will choose the root as their preferred parent, calculate their rank which is bigger than that of the root, and become part of the DODAG; then, each node in the DODAG starts to send DIO messages containing its rank. In the same way, by receiving a DIO message, each node selects a preferred parent (based on the ranks of its neighbors received through DIO messages) and specifies its rank in the DODAG. Accordingly, each node only needs to know its preferred parent to send upward messages. After choosing the preferred parent, the child node sends a DAO message to the root through its preferred parent in order to form a downward path. Afterward, it waits to receive a DAO-ACK from its parent (when the network is configured in storing mode) or from the root (when the network is configured in non-storing mode).

DODAG formation gets started from the root and gradually covers the entire network. In order to maintain the DODAG, DIO messages are periodically sent by each node based on a timer called Trickle timer. Trickle timer makes the nodes less likely to transmit the control messages while preserving network stability; in this process, as long as a node receives DIO messages compatible with its information, the node increases its Trickle-timer interval exponentially until it reaches its maximum threshold value; otherwise, it will reset to its minimum value. Besides, when a node changes its preferred parent, it sends a type of DAO message called NO-PATH to the root in order to update the downward routes. Moreover, DIS messages are used to discover neighbor nodes when there is no available node in the parent list to be selected as the preferred parent. Each node, by receiving a DIS message, either broadcasts a DIO message (In case of receiving a broadcast DIS) or resets its Trickle-timer to the minimum value (In case of receiving a unicast DIS).

3. Related work

The RPL protocol proposed in RFC-6550 cannot support mobility of the nodes properly (Oliveira and Vazao, 2016); therefore, several extensions have been proposed to improve supporting mobility in the RPL, Such as ME-RPL (El Korbi et al., 2012), MMRPL (Cobarzan et al., 2014), CO-RPL (Gaddour et al., 2014), MRPL-V (Lee et al., 2012), DMR (Hong and Choi, 2011), and mRPL (Fotouhi et al., 2015), mRPL+ (Fotouhi et al., 2017) Manikannan and Nagarajan (2020).

In MMRPL (Cobarzan et al., 2014), mobile nodes can only connect to the DODAG as leaf nodes, so they do not send any DIO messages and could not be the preferred parent for other nodes. In this protocol extension, when a mobile node chooses a preferred parent, the parent's trickle timer will be switched to a reverse Trickle-timer. The reverse Trickle-timer interval is initially set as its maximum value, and as time passes, it decreases upon its minimum value. The hypothesis is that

when a mobile node connects to a new preferred parent, the probability of remaining in the range of the preferred parent is higher at the beginning of the connection.

ME-RPL (El Korbi et al., 2012) projects the near future by the near past. It means that if a node is inconsistent in several time intervals, it will most likely be inconsistent in future intervals, which means this behavior probably represents the mobility of the node or its neighbors. Nevertheless, if a node is a bit inconsistent, its desire to maintain stability in the near future will be high. In this extension, the preferred parent changes are used as the only inconsistency parameter indicating the node's mobility within the network. Also, in this protocol, static nodes in the parent list have a higher priority than the mobile nodes to be selected as the preferred parent. In ME-RPL, the DIS interval is immediately related to the preferred parent's changes; the more the preferred parent changes, the shorter the DIS interval will be. When a node experiences high mobility, it does not wait for the next DIO period to update its parent list table, so it sends DIS messages.

The simulation results in Oliveira and Vazao (2016) show that ME-RPL and MMRPL have low responsiveness in terms of topology changes, and ME-RPL produces more control traffic with the increase in the rate of the mobile nodes, which may be due to more use of global repairs. Also, the Packet Delivery Rate (PDR) of MMRPL is less than ME-RPL and RPL.

In CO-RPL (Gaddour et al., 2014), the entire network is divided into circular regions with different radiuses, and the root is located in the center of these circles. Each of these circular regions is called a Corona, and each Corona has an ID (C-ID) sent along with other information using DIO messages. DIO receiver nodes store C_ID and LQI, which indicates the link quality between the sender and the receiver. The preferred parents will be chosen based on C_ID and LQI. Also, in this extension, a fixed timer is used to send DIO messages, and its interval will be chosen based on the speed of the mobile nodes. However, when a DIO message is received or a new neighbor is discovered, a DIO message will be sent immediately. This approach gives the protocol more responsiveness but a higher control overhead. Moreover, when a mobile node disconnects from its parent and cannot send its data upward toward the root, it will inform its children to stop sending data, and the node will send its own data to the neighbors to be forwarded upward. This approach may help the packets receive at the root but increases the communicational overhead and can be misused by malicious nodes to consume many resources of neighboring nodes.

MRPL-V (Lee et al., 2012) is specially designed for VANET. In this extension, preferred parents are chosen based on the ETX of the neighbors. A fixed value is considered for the Trickle-Timer, which is not optimal for sending DIO messages periodically. Also, in MRPL-V, each node sends a DIO message immediately after changing its preferred parent, increasing the control overhead.

According to evaluations (Oliveira and Vazao, 2016), CO-RPL and MRPL-V produce significantly more control packets than RPL, MMRPL, and MERPL, since they use a fixed Trickle-Timer, and generate control traffic based on the behavior of their neighbors, which can reach its climax in dense networks. Also, high control overhead negatively affects Packet Delivery Rate (PDR).

In DMR (Hong and Choi, 2011), hop-count is used to calculate the rank, and sometimes the LQI will also be considered to select the preferred parent. Unfortunately, DMR does not support downward routes.

In mRPL (Fotouhi et al., 2015), after transmitting a predefined number of data packets, the mobile node receives a unicast DIO message from its preferred parent, containing the Average RSSI (ARSSI) level. It will continue to transmit data until it does not receive any DIO replies or detects ARSSI degradation. Then, the mobile node initiates the discovery phase for finding a new preferred parent. In the discovery phase, the mobile node broadcasts bursts of DIS messages (more than ten DIS messages per hand-off), which significantly increases the control data overhead. Then each neighboring static receiver will send a DIO message containing the ARSSI of received DIS messages to the mobile node. Finally, the mobile node checks the received ARSSIs to find an ARSSI greater than a predefined threshold. In addition to the high communication overhead of mRPL, there is no mechanism to secure the proposed approach against malicious nodes which can continuously broadcast DIS burst messages to consume the victims' resources.

The authors of mRPL (Fotouhi et al., 2015) improved their extension by introducing mRPL+ (Fotouhi et al., 2017) which enhanced the procedure of finding proper parents by adding the ability to monitor the messages sent in the vicinity by static nodes, which decreased the hand-off delay.

Manikannan and Negarajan (Manikannan and Nagarajan, 2020) have improved mRPL (Fotouhi et al., 2015) by using an optimization algorithm called Firefly, and they reported that their new extension improved the Packet Delivery Rate by an average of 2.31% in comparison with RPL, mRPL, and mRPL+.

To put it in a nutshell, MMRPL and ME-RPL have low responsiveness but less control overhead. In contrast, the CO-RPL and MRPL-V have more responsiveness, but they produce too many control messages. Also, MRPL, MRPL+, and the improved extension provided by Manikannan and Nagarajan (2020) produce lots of control packets to manage mobility better. Furthermore, none of the surveyed studies considered the security aspects of their extension, as some particular vulnerabilities may exist in their designs, some of which were mentioned.

Accordingly, SLM-RPL is designed to maximize responsiveness while avoiding excessive control overhead. Besides, unlike other mobility extensions, security aspects and design-related vulnerabilities of SLM-RPL have been considered to make it possible to use SLM-RPL in the real-world situations. In this regard, first, an attack that can be performed on SLM-RPL called the FLI attack is introduced. Second, a probability-based method is proposed to counter DIS attacks on dynamic networks. Moreover, a lightweight IDS has been embedded in SLM-RPL, which can counter FLI attacks as well as Sybil, Rank, Sinkhole, and Impersonation attacks.

4. Proposed method

In this section, the Secured Location-Aware Mobility-enabled RPL (SLM-RPL) is proposed. There are some assumptions:

- Nodes know their current location at any moment
- The mobile nodes are considered to be leaf nodes in the network, which means the preferred parents are static nodes, like MRPL and MMRPL. Also, in MERPL, static nodes have more priority over mobile nodes to be selected as preferred parents.
- Nodes know the radius of the range covered by the preferred parent in meters (in heterogeneous networks with different devices with different ranges, the preferred parent can embed this value in DIO messages)
- Each static node, in addition to storing some information such as IP address and rank of the static neighbors, stores the location information of them in the initial network setup.

Algorithm 1periodical function in mobile nodes

1. Current Location of the Mobile node = x, y, z
2. Previous Location of the Mobile node = $x_{old}, y_{old}, z_{old}$
3. Location of the parent = $x_{parent}, y_{parent}, z_{parent}$
4. Previous Euclidean Distance from parent = d_{old}
5. Wireless range of the parent node = R
6. Current timer interval: t
7. SLM-RPL Parameters = $t_1, e_1, e_2, t_{min}, t_{inc}, t_{max}, \mu$
8. Begin
9. **If** ($|x_{old} - x| \geq e_1$ or $|y_{old} - y| \geq e_1$ or $|z_{old} - z| \geq e_1$)

(continued on next page)

(*continued*)

10.//node is mobile now
11. $d = \sqrt{(x - x_{parent})^2 + (y - y_{parent})^2 + (z - z_{parent})^2}$
12. **If** $(d - d_{old} \geq e_2)$
13.//node is moving away from its parent
14. **If** $(d - (\mu \times R) \geq e_2)$
15. **If** (There is no stored in-range parent)
16. Send_DIS ();
17. **EndIf**
18. **EndIf**
19. If $(d - R \geq e_2)$
20. **If** (There is no stored in-range parent)
21. Wait_for (t_1 seconds);
22. EndIf
23. **If** (There is at-least one stored in-range parent)
24. Change_parent ();
25. **EndIf**
26. **EndIf**
27. $t = t_{min}$
28. **EndIf**
29. Else
30. **If** $(t < t_{max})$
31. $t = t + t_{inc}$
32. **Else**
33. $t = t_{max}$
34. **EndIf**
35. EndIf
36. End.

Also, there are two main challenges for the RPL to manage mobility:

1) Detection of mobile node's disconnection from its current parent
2) Expediting the reconnection to the next proper parent.

In order to cope with the first challenge, mobile nodes should be able to calculate their distance to their current preferred parent. In this regard, each mobile node needs to know the location of its preferred parent. In SLM-RPL, static nodes embed their location information (i.e., geographic coordinates) in DIO messages. Assuming that each node knows its location at any time, nodes can calculate their Euclidian distance to their parent after receiving a DIO message from them. Suppose the distance exceeds the mobile node's wireless range configured for the node. In that case, another node should be detected and selected as the preferred parent. In this case, the mobile node broadcasts a DIS message to ask surrounding nodes to send their location information embedded in DIO messages. As a result, the mobile node can choose a new proper in-range parent as its preferred parent using the received location information.

Thereupon, according to the above description, the proposed scheme for mobility management has four building blocks, and its periodical procedures performed on mobile nodes are presented in **Algorithm 1**. The four building blocks of SLM-RPL are:

1) **Location Propagation**: Embedding geographical coordinates of static nodes in DIO messages and storing this information by the mobile receivers.
2) **Periodical Movement Detection**: Periodical movement detection and calculating the distance between mobile nodes and their preferred parents (Lines 9 to 11 of Algorithm 1).
3) **Exit Prediction**: Sending DIS message by mobile nodes before getting disconnected from their preferred parents in case of a possible exit (if no proper in-range static node has been found in the parent list) (Lines 12 to 18 of Algorithm 1).
4) **Connecting to a new parent**: Selecting a new proper in-range preferred parent (Lines 19 to 26 of Algorithm 1).

4.1. Location Propagation

Nodes can get their geographical coordinates through GPS or localization methods. In SLM-RPL, it is assumed that nodes know their location at any moment. The mobile nodes are considered leaf nodes in the network, which means the preferred parents are static nodes, like MRPL and MMRPL. Also, in MERPL, static nodes have more priority over mobile nodes to be selected as preferred parents.

In SLM-RPL, each static node first embeds its geographic coordinates in the 0 × 03 part of DIO messages, used for routing information according to the RFC-6550. The new format of the DIO base object, used for saving two bytes for each coordinate (X, Y, or Z), is shown in Fig. 1. Correspondingly, mobile nodes store the location information of static nodes after receiving a DIO.

4.2. Periodical Movement Detection

According to **Algorithm 1**, based on an adaptable periodical process, if any of the current coordinates of the mobile is changed, it means the node is moving now (Line 9 of Algorithm 1). Taking the non-optimality of the devices in the real world into account, an error parameter (e_1) is considered to validate the movement. Then, in case of a movement, the Euclidean distance between the mobile node and its parent will be calculated according to (1):

$$d = \sqrt{(x - x_{parent})^2 + (y - y_{parent})^2 + (z - z_{parent})^2} \quad (1)$$

Where x, y, and z are the current geographical coordinates of the mobile node, and x_{parent}, y_{parent}, and z_{parent} are the geographical coordinates of the preferred parent (Line 11 of Algorithm 1). Then, if the node is moving away from its parent (Line 12 of Algorithm 1), the next part of the algorithm will operate, which is presented in the next section.

Furthermore, the initial interval of the periodic timer is t_{min}, and its maximum value is t_{max}. At each timer expiration, if the mobile node is not moving away from its parent, the timer interval increases by t_{inc} up to t_{max} (Lines 29 to 34 of Algorithm 1); otherwise, the timer interval resets to t_{min} (Line 27 of Algorithm 1). This dynamic timer interval reduces the computational overhead by preventing unnecessary calculations, especially when the mobile node is not moving or not going to leave its parent's range soon.

4.3. Exit prediction

In order to expedite the connection to a new preferred parent, the mobile node needs to ask its current surrounding nodes to send their information before it leaves its parents' range. Assume that R is the radius of the range covered by the preferred parent in meters, and μ is a

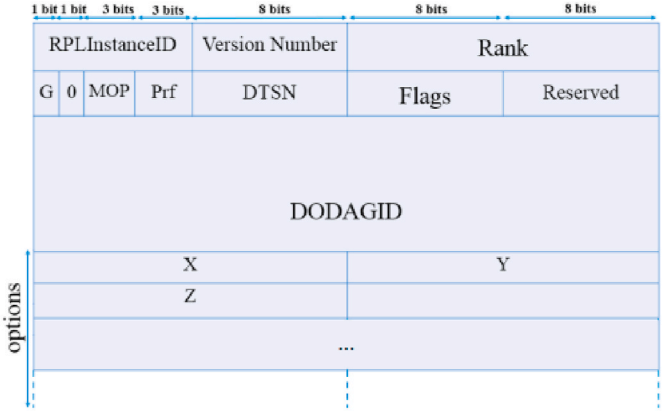


Fig. 1. DIO message format in SLM-RPL where six bytes are used for geographical information.

configurable parameter between 0 and 1. Suppose the mobile node is moving away from its parent (Line 12 of Algorithm 1), and the node distance (d) to its parent is bigger than $\mu \times R$ (Line 14 of Algorithm 1). In that case, the mobile node is more likely to leave its parent's range soon (this situation is illustrated in Fig. 2). Therefore, it checks whether there is at least one potential parent in the parent list whose current distance is less than $\mu \times R$. If such a parent does not exist, the mobile node will send a DIS message to discover proper potential parents (Lines 15 to 17 of Algorithm 1). Note that the error parameter e_2 is also considered to take the non-optimality of the devices in real-world into account in calculating the difference between distances (In lines 14 and 19 of Algorithm 1).

This approach for exit prediction aims to prevent mobile nodes from sending excessive DIS control messages. The idea behind this approach is that when a mobile node enters a new area, as there is no known parent to choose, it sends a DIS message in order to detect the static nodes in the vicinity. But afterward, if it enters that area again, it can rely on its stored information unless all of the information has expired, making it send another DIS message.

Moreover, the μ parameter can be determined based on the experiments or can be calculated dynamically according to the speed and direction of the mobile node. If μ is set to a small value, it might be too soon to be prepared for the exit; on the other hand, if μ is set to a large value, it might be late for getting prepared for the exit.

4.4. Connecting to a new parent

When a mobile node realizes it is out of the range of its parent, it has to choose a new parent. In this regard, if there is no in-range parent in the parent list, it means the mobile node has sent a DIS message recently (In line 16 of Algorithm 1), but the corresponding DIO messages are not received yet, so it should wait for t_1 milliseconds to receive the DIO messages from the neighboring nodes (Lines 20 to 22 of Algorithm 1). Afterward, if the mobile node finds at least one in-range parent in its parent list, it changes its parent (Lines 23 to 25 of Algorithm 1); otherwise, it may be out of the range of the network, and it has to try changing its parent in the next algorithm run. Moreover, in SLM-RPL, a new constraint has been defined in order to remove those parents from the parent list of a mobile node who are not in the range of the mobile node. Therefore, if there is at least one in-range potential parent in the parent list, that parent will be chosen as the preferred parent. Without this constraint, the mobile node might choose an inaccessible static node as its preferred parent, leading many packets to be lost.

5. Security considerations

RFC7416 (Tsao et al., 2015) defines RPL security requirements based on the ISO7498-2 (ISO, 1989) security reference model. These requirements include confidentiality, integrity, authentication (including data origin authentication), access control, and availability. So far, different attacks on RPL protocol have been introduced (Verma and Ranga, 2020b), and the attacks on RPL must be countered in order to address the security requirements.

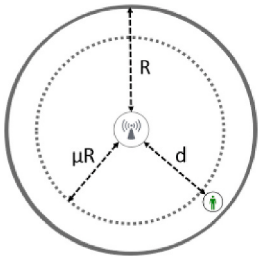


Fig. 2. An example showing a mobile node is going to leave its parent range.

Some attacks on RPL can be more hazardous than others. For instance, the Sybil attack can help the attacker bypass any IDS in IoT networks because the attacker can infinitely return to the network by simply changing his IP address after being caught. Moreover, in both rank and sinkhole attacks, the attacker can attract all neighboring nodes to select it as their preferred parents, amplifying the attacker's negative impact in performing the upcoming attacks. As a result, if the attacker could be prevented from entering the network with multiple IP addresses and attracting the neighboring nodes to select it as their preferred parents, not only his influence on the network will decrease, but also it cannot return to the network after being detected as an attacker.

Additionally, data origin authentication is one of the most important security requirements of RPL, which must be satisfied to detect Impersonation attacks in which the attackers aim to impersonate legitimate nodes.

Moreover, the announced locations by static nodes are important parts of SLM-RPL; hence the locations have to be protected against FLI attack in which the attacker falsifies the announced locations.

There have been some studies to counter different attacks on traditional static RPL networks, some of which used machine learning-based approaches (Agiollo et al., 2021; Sharma et al., 2021; Verma and Ranga, 2019; Canbalaban and Sen, 2020; Yılmaz et al., 2021), some used trust-based mechanisms (Airehrour et al., 2019; Thulasiraman and Wang, 2019; Sheibani et al., 2022), some proposed threshold-based IDSs (Guo, 2021; Almusaylim et al., 2020), and other used different approaches such as a message delivery-based method (Raza et al., 2013), a specification-based method (Le et al., 2016), a statistical-based method (Pu, 2020) and a method based on Bloom Filter and Physical Unclonable Function (Pu and Choo, 2022). However, these studies are less likely to be able to work properly in dynamic networks. The reason is that in dynamic networks extracting the normal behavior of the network is challenging, and also countering mobile attackers in a fully distributed manner may fail due to the movement of the attackers.

For example, consider a mobile attacker in an RPL network who moves around and performs DIS attacks on the surrounding nodes (i.e., broadcasting DIS messages periodically to waste the resources of victim nodes). Without mobility, the surrounding nodes could detect the attacks using a simple threshold for received DIS messages. However, in case of mobility, the normal victim nodes are less likely to detect the attacks using this approach because they probably either cannot extract the normal behavior of the system or do not have enough time to receive enough DIS messages in a time window from the attacker which moves all the time. Hence, a different approach is needed to counter this attack in dynamic networks.

Some studies tried to investigate the impact of different attacks on RPL networks, such as Dogan et al. (2022), which considers static topologies, and Aris et al. (2016), which shows the impact of Version number attack on static and dynamic RPL networks. To investigate more the impact of mobility of attackers on the destructiveness of attacks on RPL networks, DIS and Sybil attacks are simulated when attackers are static and mobile (FLI, Impersonation, and Sinkhole attacks can only be performed by static nodes as the mobile nodes are considered as leaf nodes). Here, the set D scenario in Table 3 is used in which there are 30 static nodes and eight mobile nodes (the simulation parameters are set based on Table 2). For mobile attackers, all eight mobile nodes were

Table 1
The destructiveness of static and mobile attackers on RPL.

Attack	Attacker Type	PLR %	Power Consumption overhead %	E2E delay Overhead (ms)
DIS	Static	6%	298%	1813
	Mobile	11%	365%	2054
Sybil	Static	6%	263%	611
	Mobile	8%	287%	645

considered attackers, and for static attackers, eight of the static nodes were randomly chosen as attackers. As shown in Table 1, the PLR, power consumption overhead, and E2E delay overhead caused by mobile attackers are considerably more than those caused by static attackers, meaning mobile attackers are markedly more destructive than static ones. It is to be mentioned that the used metrics will be defined in the evaluation section.

Therefore, some studies (Murali and Jamalipour, 2019; Thulasiraman and Wang, 2019; Verma and Ranga, 2020a; Medjek et al., 2015, 2017; Prathapchandran and Janani, 2021) tried to secure the RPL under mobility, but they either focused on limited attacks or did not consider the mobility extensions of RPL and their unique features.

As the DIS control messages are frequently used in SLM-RPL and most of the other RPL mobility extensions, a probability-based method is proposed to mitigate the possibility of successfully performing DIS attacks (see subsection A of this section). Additionally, unlike other mobility extensions, the security aspects and design vulnerabilities of SLM-RPL have been considered to make it possible to use SLM-RPL in the real-world situations. In this regard, an attack that can be performed on SLM-RPL called FLI is introduced in subsection B of this section.

Finally, a lightweight IDS has been proposed for static and dynamic RPL networks in subsection C of this section. As the False-Reporting attacks (i.e., sending fake attention messages, a new control message which will be introduced in subsection C of this section, by a single attacker) and Collusion attacks (i.e., sending fake attention messages by multiple attackers) can mislead IDSs in attack detection, a voting approach is considered to mitigate the possibility of successfully performing these attacks. Therefore, to put it in a nutshell, the proposed IDS can counter Sybil, Rank, Sinkhole, Impersonation, and FLI attacks, as well as False-Reporting and Collusion attacks. Also, the proposed IDS can be used to secure other RPL mobility extensions such as MMRPL (Cobarzan et al., 2014), mRPL (Fotouhi et al., 2015), mRPL+ (Fotouhi et al., 2017), and (Manikannan and Nagarajan, 2020).

5.1. A probability-based method to counter DIS attacks

DIS control messages are frequently used in RPL mobility extensions to detect surrounding nodes, but a DIS attacker can misuse these messages to consume the victims' resources.

In DIS attacks, the attacker sends lots of DIS messages to surrounding nodes. Some anomaly-based mechanisms have been proposed to detect this attack on static networks (Guo, 2021; Le et al., 2016). However, they are less likely to work correctly in most of the mobility extensions in which the normal behavior of the network is highly variable because the mobile nodes send DIS messages normally and frequently to detect new preferred parents. As a result, not only does learning the normal behavior of the network depend on the used RPL mobility extension, but also anomaly-based approaches can cause lots of false alarms in such highly dynamic networks.

In order to overcome these hurdles, a fully distributed Probability-based method is proposed here to mitigate the destructive impacts of DIS attacks. Also, the proposed method can be adapted and configured to be used in other mobility extensions for RPL.

Accordingly, the more DIS messages received from a node, the less likely it is to take the next message from that node into account. In a more detailed form, during the time window *TW* when node i receives a DIS from node j, it takes this message into account with probability P_j, and in any case, it will increase P_j using (2) (the initial value of P_j is 1):

$$P_j^{new} = \frac{1}{\theta} \times P_j^{old} \tag{2}$$

Where P_j^{new} is the newly calculated probability for considering the next DIS message from node *j*, P_j^{old} is the previously calculated one, and $\theta > 1$ is a configurable parameter.

After expiring the current time window *TW,* if N_j^{DIS} (i.e., the number of received DIS messages from node *j*) was less than or equal to τ (i.e., the number of tolerable DIS inputs), P_j increases as (3) until it reaches to 1.

$$P_j^{new} = \theta \times P_j^{old} \tag{3}$$

For example, consider the DIS attacker *k*, which sends DIS messages periodically every 5 s. When node *k* sends its first DIS, neighboring node *i* by receiving the DIS considers this message with probability $P_k = 1$, which means it definitely either broadcasts a DIO (In case of receiving a broadcast DIS) or resets its Trickle-timer to the minimum (In case of receiving a unicast DIS). Then, assuming $\theta = 2$, P_k is updated as $P_k = 1/2$, and $N_k^{DIS} = 1$. Afterward, by receiving the second DIS from the attacker node *k,* the possibility of taking that message into account is $P_k = 1/2$, and no matter what happens, the number of received DIS messages from node *k* will increase to two ($N_k^{DIS} = 2$). Assume that the time window *TW* is set to 300 s (*TW* can be varied for mobile and static neighbors since static nodes are less likely to send DIS messages frequently) and the number of tolerable DIS inputs threshold (τ) is set to 2. In this case, after *TW* expiration, N_k^{DIS} can be almost 60 (If *k* has started the attack at the beginning of *TW*), which exceeds the threshold τ, and the possibility of taking the next DIS message from node *k* into account has become $P_k = 1/2^{60}$ which is extremely small, and the attack will be ineffective.

Also, if a smart attacker node *k* wants its corresponding probability (P_k) to become reset to one by node *i* after expiring *TW*, it must send just 2 (threshold τ) DIS messages every 300 s (*TW* timer); otherwise, its corresponding probability will decrease over time. Therefore, still, the attack will be ineffective.

Finally, properly configuring τ and *TW* parameters is highly dependent on the used mobility extension. For example, in mRPL (Fotouhi et al., 2015), mRPL+ (Fotouhi et al., 2017), and Manikannan and Nagarajan's extension (Manikannan and Nagarajan, 2020), mobile nodes send more than ten DIS messages to find the new preferred parent after each hand-off; therefore, the proper τ parameter must not be less than ten. Also, the θ parameter can be configured based on the security policies considered for the network, as the larger this parameter is, the stricter it is. In SLM-RPL, the time window *TW* can be set to T_{MAX} which is the maximum length of time the RPL holds inactive neighbors' information. Accordingly, in a loss-free SLM-RPL network, static nodes can ensure that the mobile node who recently sent a DIS during this time window will not need to resend another DIS in the same time-window unless that mobile node has not found any in-range nodes to select as its preferred parent.

5.2. False-Location-Injection attack

In the False-Location-Injection (FLI) attack on SLM-RPL, the attacker is a static node that changes its real location information while sending DIO messages. As a result, mobile nodes whose preferred parents are attackers cannot calculate their correct distance from them, so they cannot correctly determine whether they are still within the range of their parents. Also, other mobile nodes may select the attacker as their preferred parent based on the received false location information.

5.3. Lightweight IDS

In this section, a lightweight IDS has been proposed for RPL networks, which can counter Sybil, Rank, Sinkhole, Impersonation, and FLI attacks as well as False-Reporting and Collusion attacks.

The static nodes can monitor the network and detect attacks as they possibly have more resources than mobile nodes. Since the structural designs of IoT networks are mostly used for the long-term (i.e., static access points are rarely repositioned or turned off after initial network setup), in a stable state, the stored information (e.g., their rank) about neighboring static nodes by them can be considered constant. In case of necessary structural changes, they can be done under the supervision of

network administrators or by using an autonomous approach to propagate eligible changes in the network structure.

Also, when a static node becomes inaccessible, some changes in ranks may happen, whose information can be propagated by the root after receiving the corresponding DAO messages sent by clients after parent change. In a more detailed form, the root can inform the neighbors of the static node whose rank changed recently by sending them unicast downward messages containing the IP address of that static node and its new rank. The static nodes in most mobility-enabled IoT applications (e.g., electronic health) have more resources than mobile nodes, so they are not likely to shut down frequently.

Using either of the mentioned approaches for handling structural changes and the mentioned approach for propagating rank changes can enable SLM-RPL, and other mobility extensions of RPL like MMRPL (Cobarzan et al., 2014), mRPL (Fotouhi et al., 2015), mRPL+ (Fotouhi et al., 2017), and Manikannan and Nagarajan's extension (Manikannan and Nagarajan, 2020) to detect attacks involved some modifications on DIO messages such as Rank, Sinkhole and FLI, and Sybil attacks.

Considering mobile nodes as leaf nodes by SLM-RPL and other mentioned extensions means only static nodes are eligible to send DIO messages. Moreover, it is assumed that each static node, in addition to storing some information such as IP address and rank of the static neighbors, stores their location information in the initial network setup (under supervision). Afterward, this information can be used to detect any abnormalities, i.e., modifications in the announced ranks, locations, or IP addresses. It is to be mentioned that, according to the addressing system were used in Contiki OS, like SVELTE (Raza et al., 2013), we stored the ID of the nodes (an unsigned integer variable) as the representative of their IPv6 addresses; however, these IDs can be easily converted to a full IPv6 address format when needed. This approach fairly decreases the memory overhead which will be investigated in the evaluation section.

Additionally, data origin authentication is one of the most important security requirements of RPL, which must be satisfied to detect attackers who aim to impersonate legitimate nodes. Here, it is assumed that a security mechanism guarantees data origin authentication, which is intended in RFC 7416; otherwise, some RSSI-based approaches can be used to validate the positions and detect impersonated nodes. Therefore, nodes can ensure that received DIO messages are coming from the claimed IP address.

Accordingly, the proposed IDS has two main parts, one part on the client nodes and another part on the root. Client nodes are responsible for detecting abnormalities and informing the root, and the root is responsible for detecting attacks based on a voting approach. In the following paragraphs, the details of the proposed IDS are provided.

On the client's side, Static client nodes monitor the DIO messages sent by other nodes. Then, after detecting an abnormality, the node will inform the root by sending a new type of RPL control message called Attention message containing the potential attacker's IP address and corresponding abnormality type.

Algorithm 2Proposed IDS function at the root

1. Received Attention-message from Node *i*: **M**
2. Number of Neighbors Of Nodes: **NN [IP_1,..,IP_N]**
3. Number of Received Attention Messages About Nodes for Different Abnormalities: **NA [IP_1,..,IP_N] [x_1,x_2,..,x_m]**
4. Begin
5. PotentialAttakcerIP←ExtractContainingIPAddrFrom (M);
6. x←ExtractContainingAbnormalityTypeFrom (M);
7. If (node *i* is a static neighbor of PotentialAttakcer and M is not duplicated)
8. NA[PotentialAttakcerIP][x] ++;
9. If (NA [PotentialAttakcerIP][x] $\geq \psi \times$ NN [PotentialAttakcerIP])
10. Raise an Attack Alarm;
11. **If (**Automatic-Informing-The-Client-Nodes is Configured**)**
12. send Attack Alarm messages to client nodes;
13. **EndIf**.

(*continued on next column*)

(*continued*)

14. **EndIf**.
15. **EndIf**.
16. **End**.

There are four types of abnormalities:

1 Receiving a DIO message from an illegitimate IP address
2 Detecting more than η (the maximum number of tolerable Sybil attacks) mobile nodes
3 Receiving a DIO message from a legitimate IP address containing modified location information
4 Receiving a DIO message from a legitimate IP address containing a modified rank

The first abnormality type is used to detect static Sybil attackers as well as illegitimate nodes who want to act as static nodes (e.g., a mobile Impersonation attacker who multicasts DIO messages). In this regard, if a DIO is received from a node whose IP address is not in the list of the eligible static neighbors, an abnormality will be considered. The second abnormality type is used to detect mobile Sybil attackers. In this regard, when a static node detects more than η (the maximum number of tolerable Sybil attacks) mobile nodes (i.e., a node with an unknown IP address is considered as a mobile node) in its vicinity, an abnormality will be considered.

The third abnormality type is used to detect FLI attacks. Based on this abnormality, if a DIO is received from a legitimate static node and the announced location in that message is different from the stored location information of that node, an abnormality will be considered. In this situation, that static node is regarded as a compromised node that has tried to falsify the announced location information to mislead its mobile child nodes in distance calculation and mobility management procedures.

Finally, the fourth type is used to detect Rank/Sinkhole attacks, in which a DIO message is received from a legitimate static node containing a different rank from one that is stored.

Also, each node is equipped with a simple firewall to isolate the attackers, based on which the attackers cannot be selected as the preferred parents, and the received control messages from them will be blocked.

On the root side, Algorithm 2 shows the part of the IDS that operates on the root. Considering that the root knows the network topology and locations of the static nodes, it obviously knows the number of neighbors of each node. Therefore, if the root receives Attention messages about node j and abnormality type x from more than ψ percent of node j's neighbors, it will raise an attack alarm (Lines 9 and 10 of Algorithm 2). Note that for mobile nodes, the average number of neighbors of static nodes will be considered as the number of neighbors. Then, based on the defined security policies, it can either inform the client automatically by sending Attack Alarm messages to client nodes (just one time per attacker) or rely on the activated alarm and let the administrators decide what to do (Lines 11 to 13 of Algorithm 2). As a result, the possibility of successfully performing of False-Reporting attacks and Collusion attacks decreases because the attackers must compromise at least ψ percent of the victim's neighbors to mislead the root.

By increasing the ψ parameter, the possibility of successfully performing Collusion attack decreases; meanwhile, the attack detection delay and false negative alarms may increase, especially when some of the sent Attention messages are lost. For example, assume $\psi = 0.9$ and the number of attacker's neighbors is 3; In this case, if one of the Attention messages is lost, the attack could not be detected at the time because $^2/_3 = 0.66 \times 100 = 66\%$ of sent Attention messages are received at the root, which is smaller than $0.9 \times 100 = 90\%$. Nevertheless, the attack can be detected sooner or later when the attacker sends more DIO messages, and none of the Attention messages is lost.

It is to be mentioned that, as we store the ID of the nodes as the

representative of the IPv6 addresses (which can be easily converted to IPv6 addresses), the memory consumption of the root for storing the topology-related information is fairly low. As it will be shown in the evaluation section, the low memory consumption overhead (about 5% RAM overhead) of the nodes in SLM-RPL enables the proposed IDS to be run on resource-constrained IoT devices.

To put it in a nutshell, according to the proposed IDS, the static clients monitor the behavior of the surrounding nodes (both static and mobile nodes), and in case of detecting an abnormality, they inform the root by sending an Attention message. This received information is used to detect Sybil, Rank, Sinkhole, and Impersonation attacks using a voting approach on the root side. Also, using the proposed voting approach mitigates the possibility of successfully performing the False-Reporting and Collusion attacks.

6. SLM-RPL simulation and evaluation

In this section, SLM-RPL is evaluated. After explaining the simulation parameters and evaluation metrics, the performance of SLM-RPL is evaluated and compared to RPL, ME-RPL, MM-RPL, MRPL-V, CO-RPL, and mRPL+. Then, the security considerations of SLM-RPL are evaluated.

In this regard, SLM-RPL is implemented using C language on the Contiki-OS and evaluated using the Cooja simulator, in which motes are emulated at the hardware level.

In order to evaluate the performance of mobility management of SLM-RPL, we used the implementation of ME-RPL, MM-RPL, MRPL-V, and CO-RPL, which were provided in the survey study conducted by Oliveira et al. (Oliveira and Vazao, 2016). Additionally, in a separated section, SLM-RPL is evaluated in the exact scenarios used in mRPL+, and the results are compared together. Finally, in the last section, the security considerations of SLM-RPL are evaluated.

6.1. Simulation parameters

Sky motes are used in the Cooja simulator, and the radio medium is configured as Unit Disk Graph Medium (UDGM). The transmission range of the nodes is set to 50 m. The network topology is rectangular, with several client nodes and a sink node almost at the top center. We used the Random Way Point (RWP) mobility model to model mobility in the network, which consists of pauses and random movements, and is used in many studies. The minimum and maximum speeds of mobile nodes are configured based on the speed of normal and fast walking of humans. Also, the accuracy of geographical coordinates is considered (<2 m) based on the real-world calculations presented by United States Federal Aviation Administration GPS Performance Analysis Report (Team, 2014). Then corresponding location error parameters (i.e., e1 and e2) are set based on the coordination's accuracy. In closing, all of the simulation parameters and corresponding values are listed in Table 2.

6.2. Evaluation metrics

Six metrics were used for the evaluations. The Packet Loss Rate (PLR) is calculated for all data packets sent in the network using (4), in which S is the number of sent packets, and R is the number of received packets at destinations.

$$PDR = \frac{S - R}{S} \tag{4}$$

The End-to-End Delay (EED) of the network is the average EED for all received data packets in the network, which is calculated using (5) (Oliveira and Vazao, 2016).

$$EED = \frac{\sum_{p=1}^{N} \left(T^{p}_{Receive} - T^{p}_{Send}\right)}{N} \tag{5}$$

Where N is the number of all received data packets in the network and p is an index for them. $T^{p}_{Receive}$ is the receiving time of the packet p in milliseconds, and T^{p}_{Send} is the sending time of the packet p in milliseconds.

Hand-off delay is the average delay for mobile nodes when they exit the range of their current parent and reconnect to a new in-range parent. Hand-off delay is calculated using (6).

$$HD = \frac{\sum_{h=1}^{M} \left(T^{h}_{Join} - T^{h}_{Leave}\right)}{M} \tag{6}$$

Where M is the number of all hand-offs that happened in the network by all mobile nodes, h is the index of each hand-off, T^{h}_{Leave} is the time when the corresponding mobile node in hand-off h leaves its parent range in milliseconds, and T^{h}_{Join} is the time when it connects to a new in-range parent in milliseconds. Also, the power consumption of the network is calculated using (7) (Raza et al., 2013).

$$\begin{aligned} Power(mW) &= \sum_{n=1}^{N} \frac{Energy(mJ)}{time(s)} = \\ &= \sum_{n=1}^{N} \frac{Energest\_Value_n \times Current \times Voltage}{RTIMER\_SECOND \times Runtime} = \\ &= \sum_{n=1}^{N} \frac{(transmit_n \times 19.5mA + listen_n \times 21.8mA + CPU_n \times 1.8 + LPM_n \times 0.0545) \times 0.33 \times 3}{4096 \times 8 \times 3600} \end{aligned} \tag{7}$$

Where N is the number of nodes in the network, n is an index for the nodes, *voltage* and *current* values are extracted from the Tmote-Sky hardware datasheet, *RTIMER_SECOND* is the number of clock ticks in each second that is also extracted from the Tmote-Sky hardware datasheet, *Runtime* is the simulation time in seconds. Moreover, $Energest\_Value_n$ represents the value obtained from the Powertrace tool in the Cooja simulator for node n when the simulation ends. It can be divided into four categories; $transmit_n$, $listen_n$, CPU_n, and LPM_n, which respectively are the number of clock ticks when node n was transmitting packets, was receiving packets, was processing, and was in the Low Power Mode (LPM).

6.3. Mobility management evaluation

In this section, first, the performance of SLM-RPL, MERPL, MMRPL, MRPL-V, and CO-RPL in mobility management is evaluated in five different parts. And then, SLM-RPL will be compared to mRPL+.

1) Evaluation of SLM-RPL on network size

This part evaluates the impact of network size on SLM-RPL and other mentioned mobility extensions. The eight different scenario sets with different network sizes are considered based on Table 3, and

Table 2
Simulation parameters.

Parameter	Value
Simulator	Cooja
Mobility model	RWP
Pause time	5 min
Min speed	1.4 m/s
Max speed	5 m/s
Mote type	Sky mote
Simulation time	3600 s
Radio medium	UDGM
Transmission range	50 m
Packet size	40 bytes
Data packet sending interval	20 s
Number of repetitions of each simulation	20
Accuracy of propagated coordination	<2 m
Error Parameter e_1	2
Error Parameter e_2	2
t_{min}	2 s
t_{inc}	2 s
t_{max}	16 s
Timer interval t_1	400 ms

The attacks are labeled as Positive, and the normal behaviors are labeled as Negative. Accordingly, TPR and FPR values are calculated using (8) and (9) (Raza et al., 2013).

$$TruePositiveRate(TPR) = \frac{TP}{TP + FN} \tag{8}$$

$$FulsePositiveRate(FPR) = \frac{FP}{FP + TN} \tag{9}$$

where *TP* is the number of true positive alarms, *TN* is the number of true-negative alarms, *FP* is the number of false-positive alarms, and *FN* is the number of false-negative alarms.

approximately 20% of the nodes are mobile. Also, the distance between static nodes is 40 m for all simulations of this part, and other simulation parameters are set according to Table 2.

According to Fig. 3.a, hand-off delay for SLM-RPL not only is dramatically lower than other protocols but almost remained constant by increasing the network size. The positive effect of this high responsiveness can be seen in Fig. 3.b, in which the PLR for SLM-RPL is much lower than others. Furthermore, according to Fig. 3.**c**, the E2E delay of SLM-RPL is much lower than CO-RPL and MRPL-V, but is higher than MMRPL, ME-RPL, and RPL. The reason might lie in the fact that almost all of the sent data packets are received in SLM-RPL, and we know that the more packets are forwarded in the network, the longer the delay will be. Also, as shown in Fig. 3.**d**, the power consumption of SLM-RPL is larger than MMRPL and MERPL, lower than CO-RPL and MRPL-V, and close to RPL, whose one of the reasons can be the more delivered packets in SLM-RPL. As a result, SLM-RPL seems to be a proper choice to be used in large loss-sensitive IoT networks.

2) Evaluation of SLM-RPL on network density

This part evaluates the impact of increasing the distance between static nodes (i.e., decreasing the network density). Four different scenarios with various distances between static nodes are considered (10, 20, 30, and 40 m). Also, all of the scenarios consist of 30 static nodes and eight mobile nodes (i.e., set D in Table 3), and other simulation parameters are set according to Table 2.

Table 3
Different scenario sets to evaluate the impact of network size.

Set	# Clients	# Static Clients	# Mobile Clients
Set A	11	9	2
Set B	20	16	4
Set C	30	24	6
Set D	38	30	8
Set E	50	40	10
Set F	60	48	12
Set G	70	56	14
Set H	80	64	16

According to Fig. 4.a, the hand-off delay for SLM-RPL is dramatically lower than other protocols, and unlike other mobility extensions, this value remains constant in both dense and sparse networks. Therefore, the PLR for SLM-RPL is also lower than others, as shown in Fig. 4.b.

Additionally, Fig. 4.c illustrates that E2E delay smoothly rises as the distance between static nodes increases because the average number of hops to the root and the hop-by-hop delay increase. Moreover, the E2E delay for SLM-RPL is close to the RPL and larger than MMRPL and MERPL because the lower PLR means a larger number of packets are forwarded and delivered, and forwarding more packets increases the end to end delay naturally.

Also, as shown in Fig. 4.d, the power consumption of SLM-RPL is close to RPL. Furthermore, on the one hand, increasing density (decreasing the distance between nodes) increases the number of each node's neighbors, which increases the power consumption of nodes due to the more number of received packets as all of the sent packets in the vicinity are received by the network card. On the other hand, increasing the density decreases the average hops towards the root because if the density were lower, the node might have to reach a node in two hops, but that node is reachable in one hop now. Accordingly, decreasing the average hops towards the root decreases the power consumption. Therefore, these two opposite impacts of increasing the density made the power consumption curve almost linear.

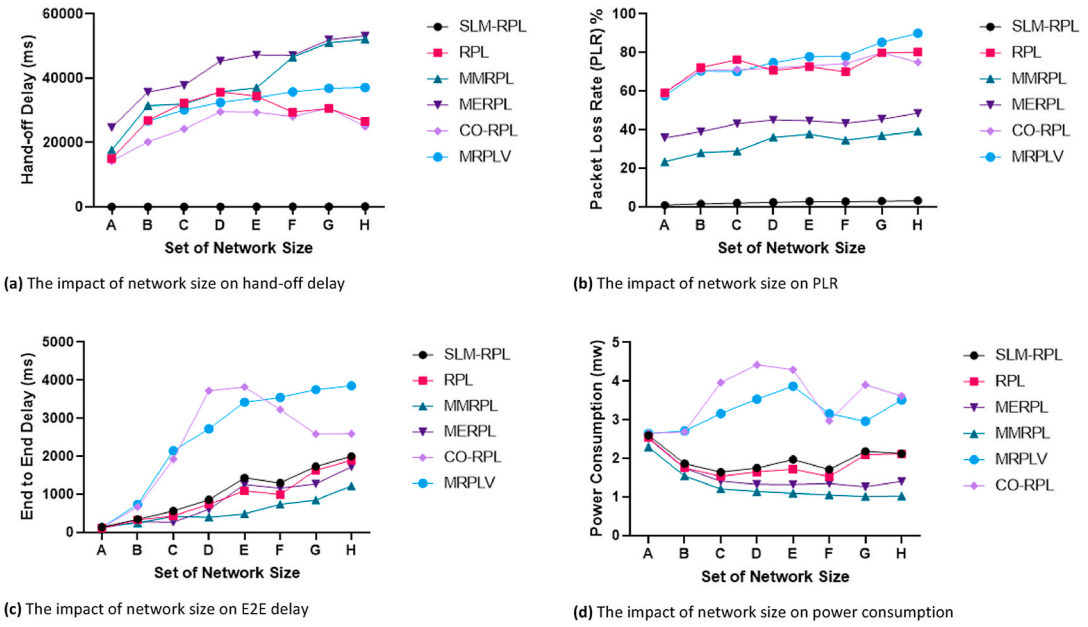
3) Evaluation of SLM-RPL on the rate of mobile nodes

This part evaluates the impact of increasing the rate of mobile nodes in the network. Five different scenarios with 38 client nodes are considered. In each scenario, the number of mobile nodes increases up to 32 (i.e., 4, 8, 16, 24, and 32 mobile nodes). Also, the distance between the static nodes is 40 m for all simulations of this part, and other simulation parameters are set according to Table 2.

According to Fig. 5.a, hand-off delay for SLM-RPL has been constantly lower than other RPL mobility extensions even by increasing the number of mobile nodes. For other extensions, increasing the number of mobile nodes (for more than eight mobile nodes) decreases the hand-off delay because with the increase of the number of mobile nodes, the ratio of static nodes to mobile nodes decreases, and given the fact that static nodes are placed at constant distances, the covered area gets smaller, and the mobile nodes are moving in a more limited area. In the same way, as shown in Fig. 5.b, the PLR for SLM-RPL is constantly and dramatically lower than other extensions, which shows the ability of SLM-RPL in mobility management in highly dynamic networks. Also, Fig. 5.c illustrates that the E2E delay of SLM-RPL by increasing the number of mobile nodes becomes better than RPL and remains larger than ME-RPL and MMRPL since more packets are delivered in SLM-RPL. Finally, Fig. 5.d shows the power consumption of SLM-RPL compared to other mobility extensions.

4) Evaluation of SLM-RPL on the number of mobile nodes

In this part, the number of static clients remains constant on 30 nodes, and the number of mobile nodes is variant in different scenarios. Also, the distance between 30 static clients is 40 m, and other simulation parameters are set according to Table 2. Keeping the number of static nodes constant and increasing the number of mobile nodes means that the network size, network density, and rate of the number of mobile nodes in the network increase simultaneously. According to Fig. 6.a, SLM-RPL still has the lowest hand-off delay among mobility extensions; therefore, it also has the lowest PLR, as shown in Fig. 6.b. Also, E2E delay and power consumption for SLM-RPL remained close to the RPL even in highly dynamic networks, as shown in Fig. 6c and d.



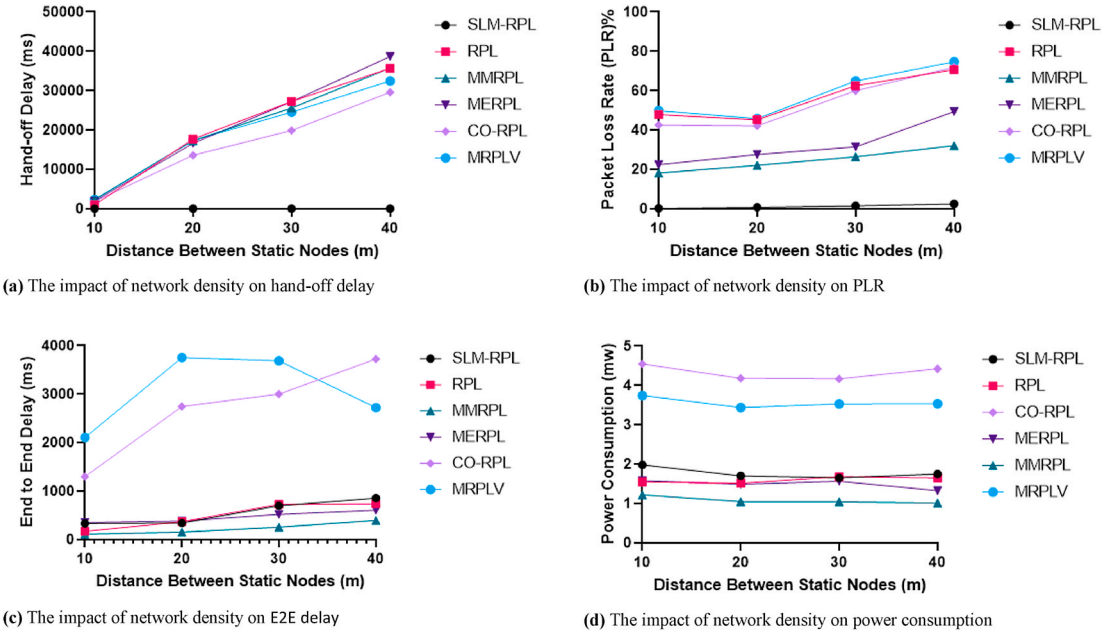
(a) The impact of network size on hand-off delay

(b) The impact of network size on PLR

(c) The impact of network size on E2E delay

(d) The impact of network size on power consumption

Fig. 3. The impact of network size on SLM-RPL and related methods.



(a) The impact of network density on hand-off delay

(b) The impact of network density on PLR

(c) The impact of network density on E2E delay

(d) The impact of network density on power consumption

Fig. 4. The impact of network density on SLM-RPL and related methods.

According to the results of the previous sections, in all scenarios, the PLR and hand-off delay for SLM-RPL were dramatically better than other extensions, but the EED and power consumption for SLM-RPL have a moderate position between other RPL extensions. Therefore, to compare SLM-RPL with other protocols based on different requirements of IoT applications, a fusion formula in (10) is provided, dividing evaluation metrics into two groups and assigning weights to them. Previous sections show that PDR and hand-off delay have a direct relationship; therefore, they have been considered together as a separate group in (10). Also, this approach helps us investigate which of the RPL extensions has the best performance to be used in loss-sensitive IoT applications.

$$f(PLR, HD, EED, Power) = \alpha(PLR + HD) + (1-\alpha)(EED + Power) \tag{10}$$

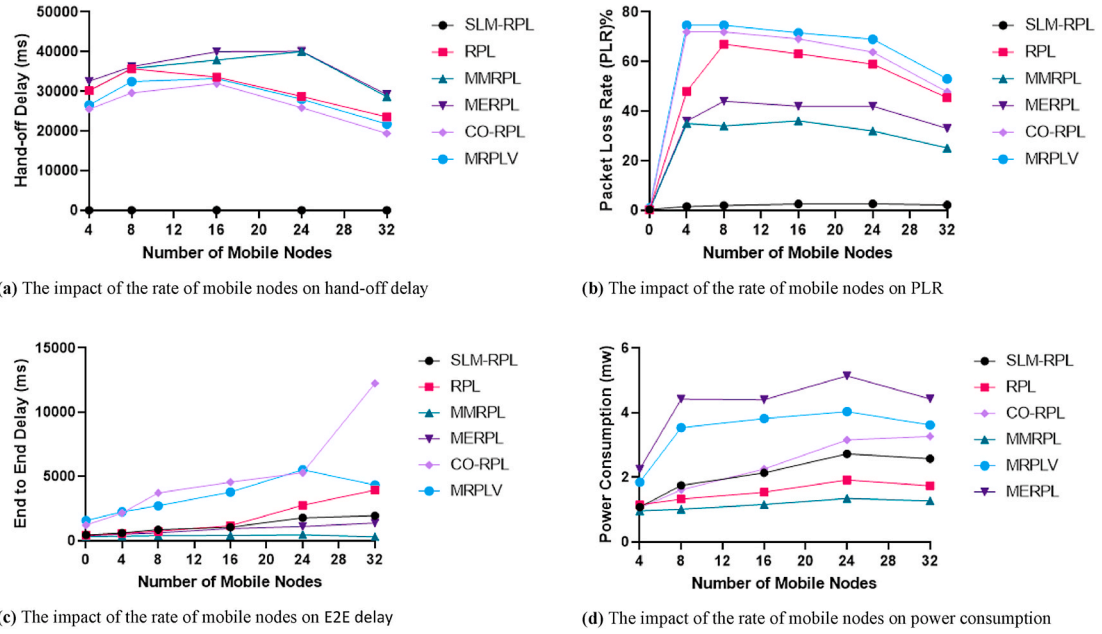
where α is the coefficient of influence (i.e., the weight).

In this regard, three different values for α are considered; 0.25, 0.5, and 0.75; then, a separate chart is presented for each of these values. In this section, the simulation results of the previous section (section 4) are considered, and the obtained values for evaluation metrics are normalized using (11).

$$x' = \frac{x - \min}{\max - \min} \tag{11}$$

where x' is the normalized value of x, min is the minimum value of x, and max is the maximum value of x.

Fig. 7.a illustrates the results of the fusion formula for $\alpha = 0.25$, which means the power consumption and E2E delay are more important than PLR and hand-off delay. In this situation, MMRPL seems to be the



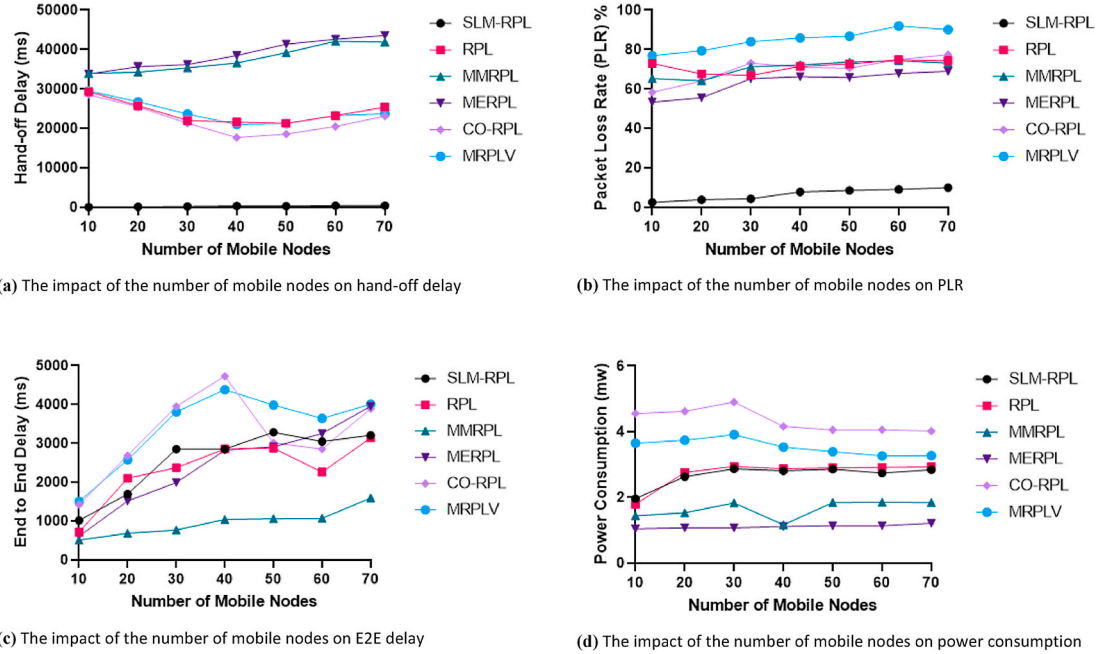
(a) The impact of the rate of mobile nodes on hand-off delay

(b) The impact of the rate of mobile nodes on PLR

(c) The impact of the rate of mobile nodes on E2E delay

(d) The impact of the rate of mobile nodes on power consumption

Fig. 5. The impact of the rate of mobile nodes on SLM-RPL and related methods.



(a) The impact of the number of mobile nodes on hand-off delay

(b) The impact of the number of mobile nodes on PLR

(c) The impact of the number of mobile nodes on E2E delay

(d) The impact of the number of mobile nodes on power consumption

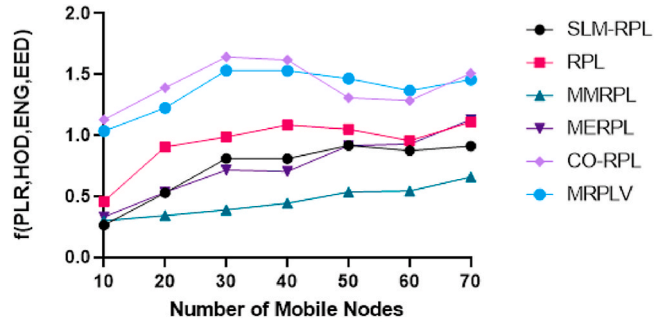
Fig. 6. The impact of the number of mobile nodes on SLM-RPL and related methods 5) Evaluation of SLM-RPL on a fusion of metrics.

best option. Fig. 7.b illustrates the results of the fusion formula for $\alpha = 0.5$, which means all evaluation metrics have equal importance. In this situation, SLM-RPL seems to be the best option. Finally, Fig. 7.c shows the results of the fusion formula for $\alpha = 0.75$, which means PLR and hand-off delay have more importance than power consumption and E2E delay. In this situation, SLM-RPL seems to be the best option.

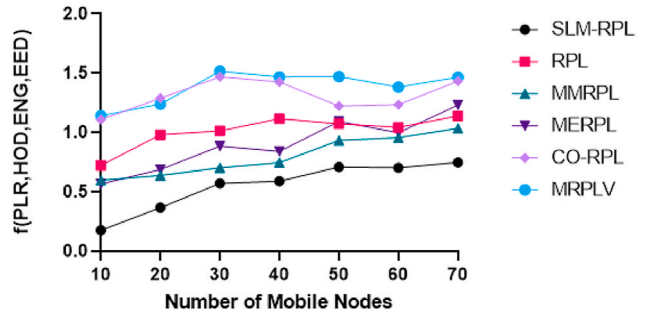
6) Comparison with mRPL+

As described in Related Work, mRPL+ (Fotouhi et al., 2017) is the enhanced version of mRPL. The evaluation metrics used in that study are hand-off delay, total packet overhead, and packet delivery rate (PDR). Surprisingly, in the mRPL + study, in all of the simulations, the simulation duration has been set to just 2 min, which is too short for accurately evaluating a mobility extension. However, we tried to simulate the exact scenarios used in that study. In this regard, they simulated one mobile node and 12 static nodes in an $8\,m \times 12\,m$ room. The mobile node pauses for 30 s firstly and then moves in a defined route with a speed interval [0.5 m/s, 2 m/s]. In addition, they have studied the impact of different data transmission periods (0.05, 0.1, 0.5, 1, 2, and 5 s) on evaluation metrics.

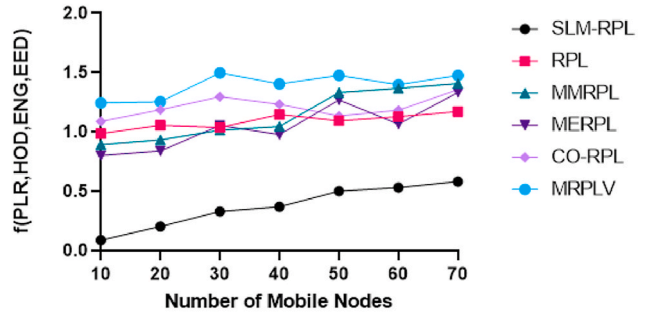
According to Fig. 8.a, in extreme situations in which the data transmission period is less than 1 s, the hand-off delay of mRPL+ is less than SLM-RPL, but for other states, the hand-off delay of SLM-RPL is dramatically lower. Also, the hand-off delay for SLM-RPL has almost remained constant. Moreover, as shown in Fig. 8.b, the PLR for mRPL+



(a) Comparison using fusion formula with α=0.25

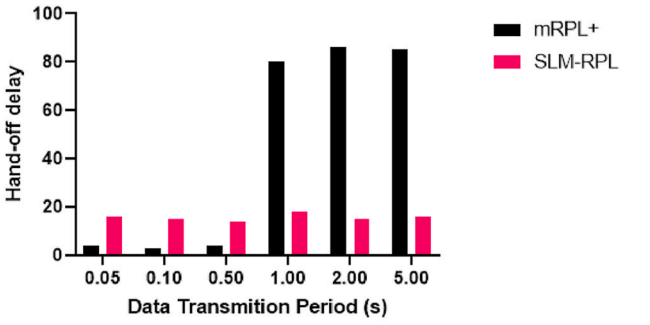


(b) Comparison using fusion formula with α=0.5

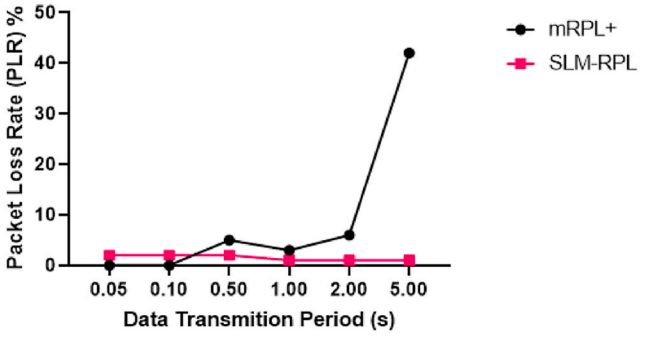


(c) Comparison using fusion formula with α=0.75

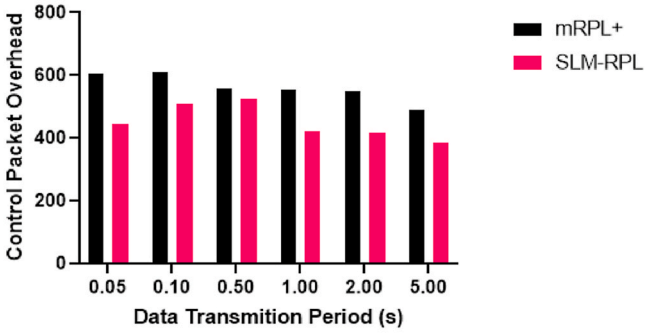
Fig. 7. Comparison between SLM-RPL and related methods based on the fusion formula.



(a) Comparison between hand-off delay of SLM-RPL and mRPL+



(b) Comparison between PLR of SLM-RPL and mRPL+



(c) Comparison between control overhead of SLM-RPL and mRPL+

Fig. 8. Comparison between SLM-RPL and mRPL+.

in extreme networks is slightly less than SLM-RPL, nevertheless, for other states, the PLR of SLM-RPL becomes less, and for the 5-s data transmission period, the PLR for SLM-RPL remains close to 0, but that of mRPL + has ascended to 42%. Finally, Fig. 8.c shows that the number of control packets sent in mRPL+ was more than that of SLM-RPL. Note that for the first couple of minutes, the RPL network is not stable yet, so in 2 min simulations, the number of sent control packets is high naturally. As a result, if the simulations have been run for longer durations, the difference between the number of sent control packets of two protocols could be even more, as the mRPL + sends some burst of control messages during the hand-off process.

7) Discussion on mobility results

The previous sections show that in all scenarios, SLM-RPL dramatically outperformed the other evaluated extensions in terms of hand-off delay and PLR and has a moderate position in terms of E2E delay and power consumption.

The reason for the brilliant results in hand-off delay and, consequently, PLR is the fact that SLM-RPL can effectively deal with the two main challenges mentioned at the beginning of section IV: detection of mobile node's disconnection from its current parent and expediting the reconnection to the next proper parent.

Mobile nodes in SLM-RPL periodically calculate their distance to their preferred parent, and in case of detecting a possible disconnection, they prepare themselves by finding new in-range parents before being disconnected. Therefore, when the mobile node detects that it has been out of the range of its parent, it can easily reconnect to another in-range parent, which reduces the hand-off delay to a great deal.

Also, using a fusion formula, it has been shown that not only does SLM-RPL outperform other extensions in loss-sensitive applications (in which the hand-off delay and PLR are more important than E2E delay and power consumption), but also it has the best results among other extensions when the importance of all metrics is equal.

In this section, the security considerations of SLM-RPL are evaluated. In this regard, first, the performance of the proposed IDS is evaluated using a variety of scenarios; then, it is compared to recent related work; finally, the proposed probability-based method for mitigating the impact of DIS attacks is evaluated.

In all sections, the set D scenario introduced in Table 3 is considered

in which there are 30 static client nodes, eight mobile client nodes, and a root node. Also, the distance between static nodes is 30 m, and other simulation parameters are set based on Table 2.

1) Evaluation of the proposed IDS

This section aims to evaluate the performance of the proposed IDS in detecting FLI, Sybil, impersonation, and rank/sinkhole attacks. In order to evaluate the proposed IDS comprehensively and to simulate real-world situations, a variety of parameters are evaluated: ψ parameter, which is proposed in Algorithm 2, error probability (i.e., the probability of a packet not being received successfully), and the rate of attackers. Moreover, the impact of the η parameter (i.e., the maximum number of tolerable Sybil attacks) and Sybil attack interval are also evaluated for Sybil attacks.

For all attacks, six static nodes, which are randomly chosen, perform collusion attacks on their neighbors by sending fake Alarm messages about them to the root per second. Also, three different cases with different values for the ψ parameter (i.e., 0.25, 0.5, and 0.75) are considered to evaluate the impact of this parameter. Three different cases with different attacker rates (i.e., 10% and 20%, and 30% of nodes) are also used to evaluate the impact of increasing the rate of attackers. Plus, two different cases with different error probabilities (10% and 20%) are used to evaluate the impact of this parameter.

Moreover, for Sybil attacks, three different cases are considered to simulate different Sybil attacks in terms of attack interval: extreme Sybil attacks (i.e., 1-s interval), moderate Sybil attacks (i.e., 10 s interval), and less-frequent Sybil attacks (i.e., 30 s interval). Finally, the η parameter is set to 10, 20, or 30 to be evaluated.

a) Evaluating SLM-RPL under FLI attack

For evaluating the impact of the ψ parameter under FLI attacks, approximately 10% of nodes (i.e., four static nodes out of 38 nodes) are considered as FLI attackers, and the error probability is set to 10%. As shown in Fig. 9.a, for all ψ values, TPR and Accuracy remain close to 1. Plus, FPR for $\psi = 0.25$ is 0.021, showing that some normal behaviors are mistakenly considered as attacks due to collusion attacks. However, FPR for $\psi = 0.5$ and $\psi = 0.75$ is zero because the collusion attacks could not be performed successfully since more than 50% of attackers should be compromised to make it possible. Finally, TPR for $\psi = 0.5$ is slightly higher than $\psi = 0.75$ because in the latest case, the root could not detect some of the attacks since the quorum is not reached (i.e., some of the required sent Attention messages are lost when there is an error probability of 10%). As a result, the best value for ψ parameter seems to be 0.5.

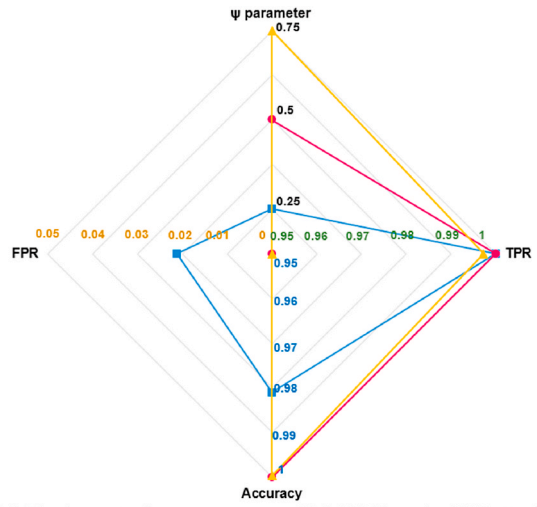
For evaluating the impact of the rate of attackers, the ψ parameter is set to 0.5, and the error probability is considered to be 10%. According to Fig. 9.b, even by increasing the rate of attackers to almost 30% (i.e., 12 static nodes), FPR remains close to zero, and TPR, as well as accuracy, remains close to 1, which means it does not decrease the efficiency of the proposed IDS in detecting FLI attacks.

Finally, for evaluating the impact of increasing the error probability, the ψ parameter is set to 0.5, and the rate of attackers is considered to be nearly 10% (i.e., four static nodes). As shown in Fig. 9.c, doubling the error probability to 20% slightly decreases TPR and Accuracy, but they are still close to 1. Also, FPR remains intact by increasing the rate of attackers.

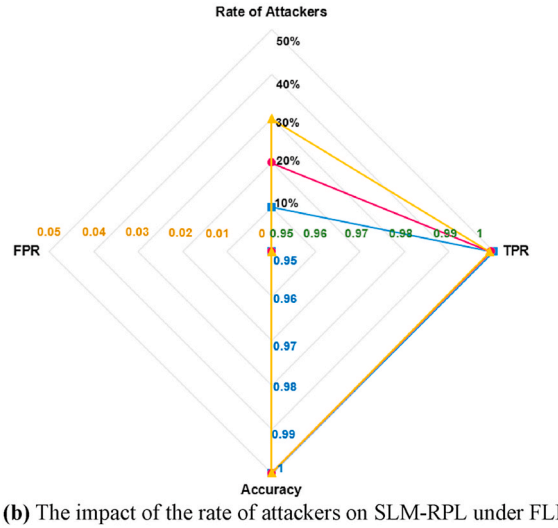
b) Evaluating SLM-RPL under Rank/Sinkhole attack

As has been said before, Rank and Sinkhole attacks are the same in the context of RPL, so in this section, both of them are evaluated together.

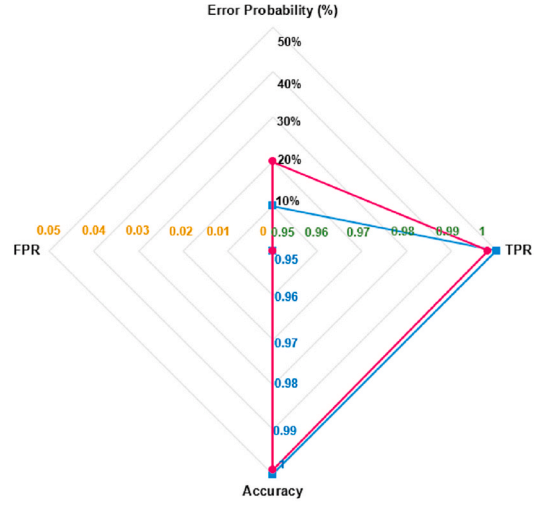
For evaluating the impact of the ψ parameter under Rank/Sinkhole attacks, almost 10% of nodes (i.e., four static nodes) are considered as



(a) The impact of ψ parameter on SLM-RPL under FLI attack

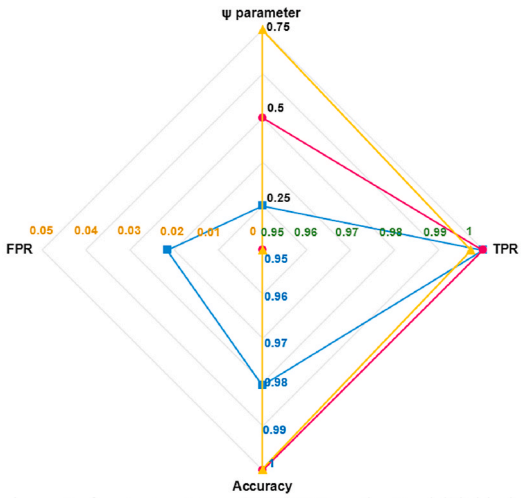


(b) The impact of the rate of attackers on SLM-RPL under FLI attack

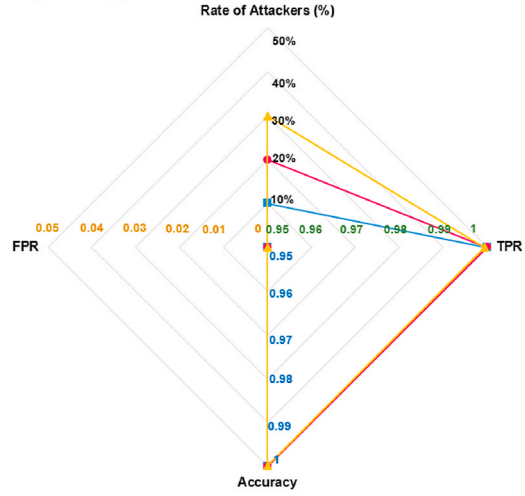


(c) The impact of error probability on SLM-RPL under FLI attack

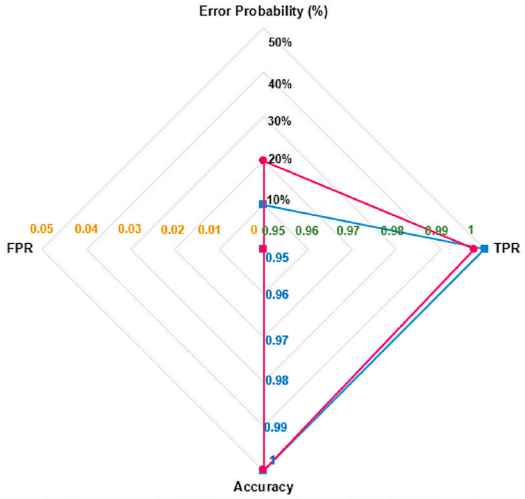
Fig. 9. Evaluating SLM-RPL under FLI attack.



(a) The impact of ψ parameter on SLM-RPL under Rank/Sinkhole attack



(b) The impact of the rate of attackers on SLM-RPL under Rank/Sinkhole attack



(c) The impact of error probability parameter on SLM-RPL under Rank/Sinkhole attack

Fig. 10. Evaluation of SLM-RPL under Rank/Sinkhole attack.

Rank/sinkhole attackers, and the error probability is set to 10%. As shown in Fig. 10.a, as in the previous section, for all ψ values, TPR and Accuracy remain close to 1, and FPR for $\psi = 0.25$ is approximately 0.022, which shows that some of the normal behaviors are mistakenly considered as attacks as a result of collusion attacks. Also, the results for $\psi = 0.5$ are better than others because it is neither too small nor too big.

To evaluate the impact of increasing the rate of attackers, the ψ parameter is set to 0.5, and the error probability is considered 10%. According to Fig. 10.b, even by increasing the rate of attackers to nearly 30% of nodes, FPR remains close to zero, and TPR and Accuracy remain close to 1, which means that the proposed IDS can still detect attacks efficiently.

Finally, for evaluating the impact of increasing the error probability, the ψ parameter is set to 0.5, and the rate of attackers is considered to be almost 10% of nodes. As shown in Fig. 10.c, increasing the error probability to 20% slightly reduces TPR and Accuracy; however, they remain close to 1, and FPR also remains zero.

c) Evaluating SLM-RPL under Impersonation attack

In this section, to evaluate the impact of the ψ parameter, 10% of nodes (i.e., four mobile nodes) are considered as impersonation attackers, and the error probability is set to 10%. According to Fig. 11.a, similar to what has been discussed for the other two previous attacks, the case with $\psi = 0.5$ has the best results because it is neither too small that collusion attacks can be performed successfully nor too big that the lost Alarm messages cause not detecting the attacks at the root.

To evaluate the impact of increasing the rate of attackers, the ψ parameter is set to 0.5, and the error probability is considered 10%. According to Fig. 11.b, even by increasing the rate of attackers to approximately 20% of the nodes (i.e., all of the eight mobile nodes are attackers), FPR remains to zero, and TPR and Accuracy remain close to 1, which means it does not decrease the efficiency of the proposed IDS in detecting Impersonation attacks. Finally, for evaluating the impact of increasing the error probability, the ψ parameter is set to 0.5, and the rate of attackers is considered to be almost 10%. As shown in Fig. 11.c, although doubling the error probability to nearly 20% decreases TPR and Accuracy slightly, they remain close to 1. Moreover, FPR remains intact by increasing the rate of attackers.

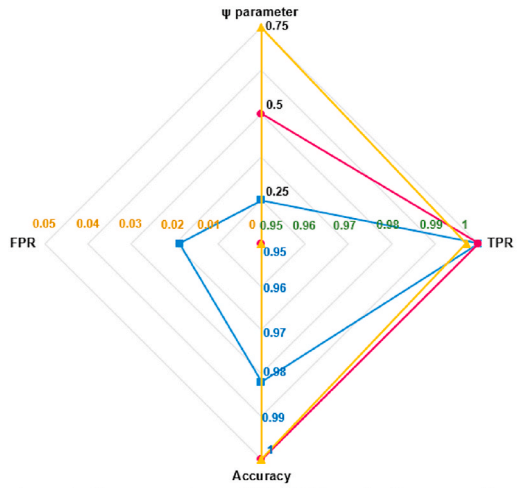
To evaluate the ψ parameter, approximately 10% of nodes are considered as Sybil attackers (i.e., two static and two mobile attackers), and the error probability is set to 10%. According to Fig. 12.a, in all cases, TPR is higher than 0.98, and accuracy for $\psi = 0.25$ and $\psi = 0.5$ is close to 1. Also, FPR for the scenario with $\psi = 0.25$ is 0.038 because, in this case, the attackers have successfully performed some of the collusion attacks since fewer false alarm messages are needed to mislead the root node.

For evaluating the impact of the rate of attackers, the ψ parameter is considered to be 0.5, and the error probability is set to 10%. As shown in Fig. 12.b, increasing the rate of attackers to nearly 20% of nodes (i.e., four static and four mobile Sybil attackers) slightly decreases TPR and Accuracy; however, TPR remains higher than 0.98, and accuracy remains higher than 0.99.

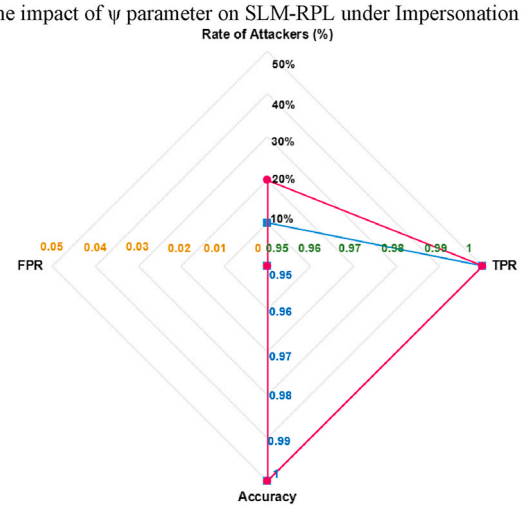
To evaluate the impact of the error probability, the ψ parameter is set to 0.5, and the rate of attackers is considered to be nearly 10% of nodes. Fig. 12.c shows that doubling the error probability slightly reduces TPR and accuracy; however, TPR remains higher than 0.98, and accuracy remains close to 1.

Furthermore, to evaluate the impact of the maximum number of tolerable Sybil attacks (i.e., η parameter), the ψ parameter is set to 0.5, the rate of attackers is considered to be nearly 10% of nodes, and the error probability is set to 10%. According to Fig. 12.d, increasing the maximum number of tolerable Sybil attacks decreases TPR slightly because more Sybil attacks are not detected before the maximum is reached.

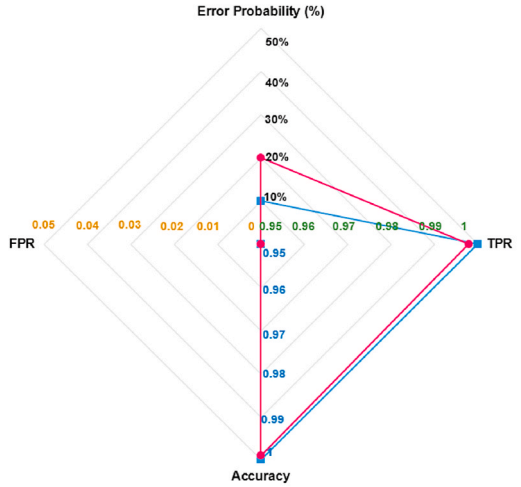
Finally, to evaluate the impact of the Sybil attack interval, the η



(a) The impact of ψ parameter on SLM-RPL under Impersonation attack



(b) The impact of the rate of attackers on SLM-RPL under Impersonation attack



(c) The impact of error probability on SLM-RPL under Impersonation attack

Fig. 11. Evaluation of SLM-RPL under Impersonation attack.

parameter is set to 10, the ψ parameter is set to 0.5, the rate of attackers is considered to be nearly 10% of nodes, and the error probability is set to 10%. According to Fig. 12.e, in an extreme situation when the interval is 1 s, SLM-RPL can detect Sybil attacks with TPR and Accuracy close to 1 and FPR = 0. Also, in a moderate situation, when the interval is 10 s, accuracy remains close to 1, and TPR is higher than 0.98. Finally, increasing the interval of Sybil attack to 30 s decreases TPR to almost 0.96, but accuracy remains near 1. These results show that the proposed IDS can detect extreme and moderate Sybil attacks more effectively than less frequent ones. Still, accuracy for all cases is close to 1, and FPR is zero.

e) Discussion on the results in terms of performance and power consumption

Based on the analysis of the results obtained from previous sections, Table 4 shows the best result and the average of results for each attack, in which the average of all of the simulations related to that attack is used to calculate the average values. For all attacks, the best result is achieved when 10% of the network are attackers, the error probability is 10%, and the ψ parameter is set to 0.5. Furthermore, the best result for the Sybil attack is for the extreme scenario in which the attackers change their IP address per second, and the maximum number of tolerable Sybil attacks for the best scenario is 10. Finally, the average power consumption for each attack is shown in the Table.

f) Comparison with related work

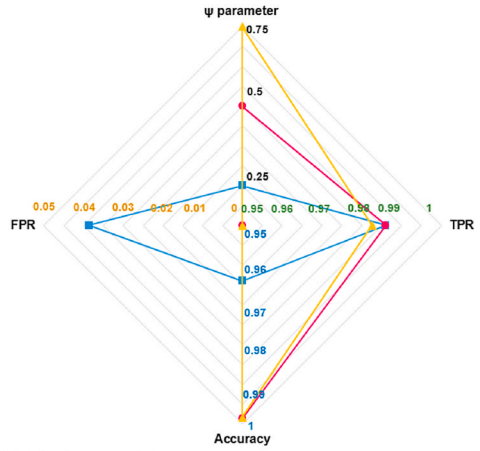
First, SLM-RPL is simulated and evaluated in the same scenarios presented in Murali and Jamalipour (2019), in which three different scenarios for Sybil attack are proposed, and an IDS is provided to detect them. In the first scenario, Sybil attackers are static and target one fixed region. In the second scenario, the attackers are fixed but scattered in the network. Finally, in the third attack scenario, Sybil nodes are under mobility and distributed among the network. For all scenarios, 80 nodes are randomly placed into a 300 m × 300 m area. Note that the exact topology was not provided in Murali and Jamalipour (2019) so we used a random topology to simulate SLM-RPL. simulation time is 3000 s, and also a main Sybil attacker is considered, and the Sybil attack ratio during simulations varies from 0.1 to 0.6. As shown in Table 5, SLM-RPL has achieved significantly better results in all three scenarios, especially in scenario 3 when attackers are mobile, which shows that the proposed IDS can detect Sybil attacks in dynamic environments more efficiently.

Secondly, SLM-RPL is compared to Prathapchandran and Janani (2021) in terms of Sinkhole attack detection based on the scenarios used in that study, which aimed to propose a trust-aware method to counter Sinkhole attacks on RPL. In this regard, different scenarios with different rates of attackers are used, in which 100 static nodes are placed into a 300 m × 300 m area in the Cooja simulator. Note that we randomly placed nodes in the network since the exact topology was not provided in Prathapchandran and Janani (2021). The time of simulation is 3600 s. Results for some scenarios were partially available in that study; however, using the available results, according to Table 6, SLM-RPL can detect Sinkhole attacks with higher accuracy and lower FPR.

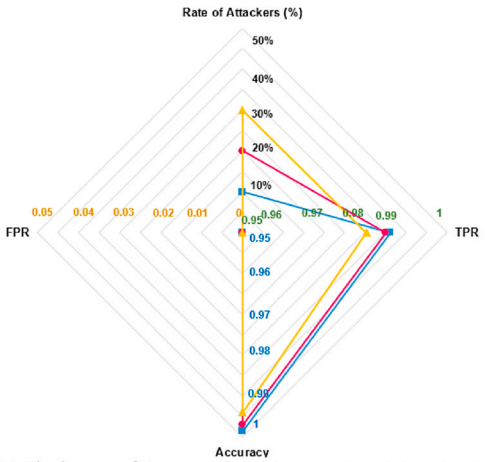
(2)Evaluation of proposed Probability-based method for countering DIS attacks.

This section evaluates the proposed probability-based method for countering DIS attacks.

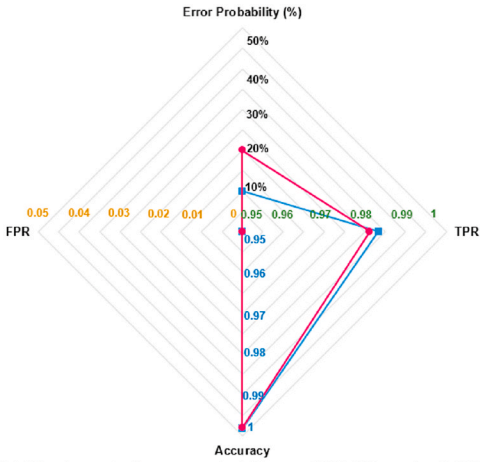
Finding proper values for parameters of the proposed method depends on the characteristic of the network, such as the used extension and the speed of the mobile nodes. Here, scenario D in Table 3 is used, involving 30 static nodes, eight mobile nodes, and a root node. Other parameters are set based on Table 2. After simulating different networks in the absence of attackers, based on the observations, static nodes send a maximum of one DIS message every 15 min (900 s), and mobile nodes send a maximum of one DIS message every 5 s; therefore, the *TW* for static and mobile neighbors are respectively considered to be 900 and 5 s. Also, *the* τ parameter is set to one since we are almost sure that in each *TW,* only one DIS message is sent by normal nodes. Also, it was observed that assigning a value higher than 2 for the θ parameter would not



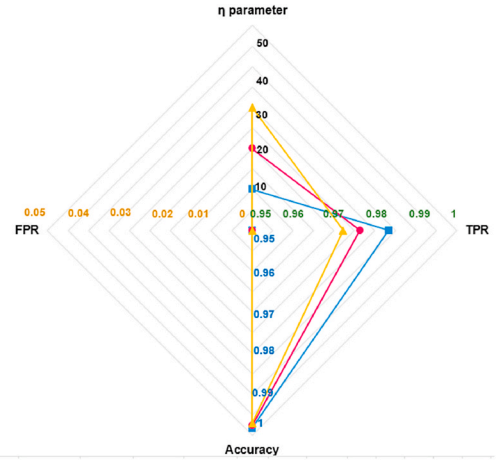
(a) The impact of ψ parameter on SLM-RPL under Sybil attack



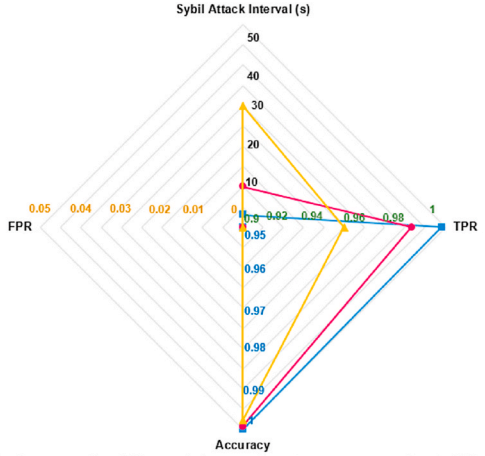
(b) The impact of the rate of attackers on SLM-RPL under Sybil attack



(c) The impact of error probability on SLM-RPL under Sybil attack



(d) The impact of η parameter on SLM-RPL under Sybil attack



(e) The impact of Sybil attack interval on SLM-RPL under Sybil attack

Fig. 12. Evaluation of SLM-RPL under Sybil attack.

change the results; therefore, the θ parameter is set to 2.

Since DIS attacks cause victim nodes to send DIO messages, the most accurate approach to evaluate the performance of the proposed probability-based method is to study the total number of sent DIO messages. In this regard, for each scenario, first, in the presence of attackers, the proposed method is disabled, and the number of sent DIO messages is counted. Second, the number of sent DIO messages is counted when the proposed method is enabled. Finally, these two numbers are reported along with the number of sent DIO messages when there are no attackers.

In order to evaluate the performance of SLM-RPL in detecting different DIS attacks, the impact of two different parameters is evaluated: the type of DIS messages sent by attackers (i.e., unicast or multicast), and the DIS attack interval (i.e., 0.01, 0.05, 1, 1.5, 2, 10, and 20 s). Also, nearly 20% of nodes are considered to be DIS attackers (four static and four mobile attackers).

Table 4
Best and average Results for different attacks.

Attack		TPR	Accuracy	FPR	Power Consumption (mw)
FLI	Best	1	1	0	1.869123
	Average	0.9999	0.9975	0.0026	1.88588
Rank/ Sinkhole	Best	1	1	0	1.89
	Average	0.9993	0.9974	0.0026	1.907081
Impersonation	Best	1	1	0	1.868707
	Average	0.9992	0.9977	0.0023	1.86903
Sybil	Best	0.9983	0.9998	0	1.922254
	Average	0.981	0.9952	0.0028	1.867978

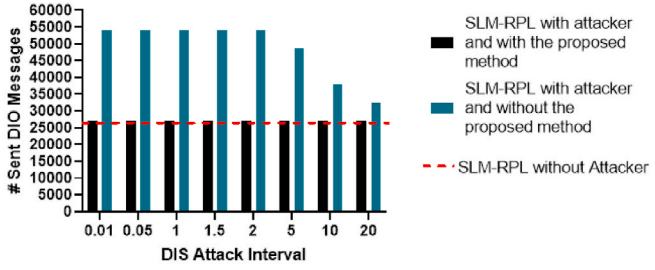
In this section, the performance of the proposed IDS under Sybil attacks is evaluated.

Table 5
Comparison between slm-rpl and Murali and Jamalipour (2019) in Sybil attack detection.

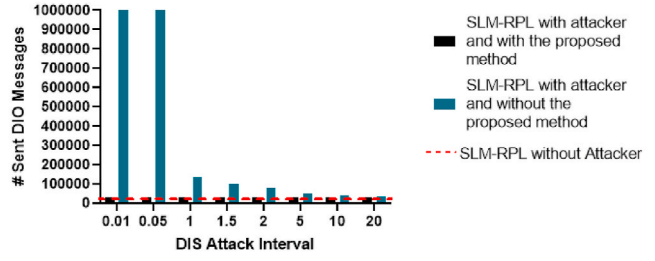
Scenario #	TPR	TPR	Accuracy	FPR
1	Murali and Jamalipour (2019)	0.974	0.97	0.048
	SLM-RPL	0.9875	0.993	0
2	Murali and Jamalipour (2019)	0.935	0.952	0.096
	SLM-RPL	0.986	0.992	0
3	Murali and Jamalipour (2019)	0.955	0.948	0.148
	SLM-RPL	0.983	0.991	0

Fig. 13.a and Fig. 13.b show the total number of sent DIO messages in the network, respectively, when attackers send unicast and multicast DIS messages. According to both of them, the number of sent DIO messages when the proposed method is used is way lower than when it is disabled. The result for the proposed method is also close to the "SLM-RPL without attacks" which is shown with a red dashed line. Also, the difference between "SLM-RPL without the proposed method" and "SLM-RPL without attack" is higher in Fig. 13.b because when the DIS messages are sent with multicast address, the neighboring nodes reactively respond with a DIO message regardless of the Trickle timer, but in Fig. 13.a, when DIS messages are unicast, the victims merely reset their Trickle Timer; hence, the interval of sending DIO messages in this situation is limited to the minimum value of Trickle timer (here, it is set to 2 s which is the default value in Contiki-OS).

More accurately, for unicast DIS attacks, "SLM-RPL without the proposed method" sends almost 80% more DIO messages than "SLM-RPL without attack" on average for all intervals; however, that for "SLM-RPL with the proposed method" is less than 0.2%. Moreover, for multicast DIS attacks, "SLM-RPL with the proposed method" has less than 0.02% overhead in sending DIO messages; in contrast, the "SLM-RPL without the proposed method" sends 6125% more DIO messages than "SLM-RPL without attack". As a result, SLM-RPL can efficiently counter both types of DIS attacks with different attack intervals.



(a) The impact of different attack intervals on the total number of DIO messages sent in SLM-RPL under unicast DIS attacks



(b) The impact of different attack intervals on the total number of DIO messages sent in SLM-RPL under multicast DIS attacks

Fig. 13. Evaluating the proposed method for countering DIS attacks.

E Memory consumption

Table 7 shows the extra ROM requirements of SLM-RPL in different configurations, which is well below the total available ROM in constrained devices such as 48k in Tmote sky. Accordingly, the mobile nodes' ROM overhead compared to RPL is 260 bytes (0.58% overhead), which means that SLM-RPL mobile nodes require about 0.5% of available ROM on Tmote Sky motes. Also, the ROM requirement for static nodes and the root node in SLM-RPL are 402 and 548, respectively. Table 8 shows the maximum RAM usage of SLM-RPL with different configurations (based on the results obtained from the simulations in previous sections), based on which the mobile nodes, static nodes, and the root node require a maximum of 210 bytes, 279 bytes, and 554 bytes, respectively. It is to be noted that the total RAM for Tmote Sky motes is 10k, so although most of the researchers assume that the root node has more memory capacity than others, SLM-RPL requires a maximum of 548 bytes at maximum on the root node, which means it consumes 5% of Tmote Sky motes at the worst case. In summary, the results show that SLM-RPL can easily run on resource-constrained IoT devices.

7. Threats to validity

SLM-RPL is shown to be able to manage mobility on IoT networks properly. However, some conditions may negatively affect the performance of SLM-RPL:

1- Highly inaccurate location data: the mobile nodes periodically calculate their distance to their parent by using the location information. It has been assumed that the mobile nodes know their current location at any time. The non-optimality of the real-world situations is considered in SLM-RPL by using error parameters in calculations. However, the performance of SLM-RPL may decrease in case of high errors in the location data.

2- As it has been shown in the evaluation section, if for an application the power consumption and E2E delay are more important than hand-off delay and PLR, MMRPL would be a better option than SLM-RPL. However, by customizing the parameters of SLM-RPL (e.g., the value of timers used for periodical operations of SLM-RPL), we might be able to make a trade-off between these different criteria, which can be investigated in future works.

3- Like some other mobility extensions (e.g., MRPL and MMRPL, and

Table 6
Comparison between slm-rpl and prathapchandran and janani (Prathapchandran and Janani, 2021) in sinkhole attack detection.

Attacker Rate (%)	TPR	TPR	FPR
10	Prathapchandran and Janani (2021)	–	0.183
	SLM-RPL	1	0
20	Prathapchandran and Janani (2021)	–	0.279
	SLM-RPL	0.9995	0
30	Prathapchandran and Janani (2021)	–	–
	SLM-RPL	0.9912	0
40	Prathapchandran and Janani (2021)	–	–

Table 7
ROM usage of slm-rpl in different configurations.

Node Type	Configuration	ROM (byte)	Overhead (byte)
Mobile Clients	**Mobility**	44,958	214
	IDS Firewall	44,790	46
	Total	**45,004**	**260**
Static Clients	**Mobility**	44,836	92
	DIS Method	44,870	126
	IDS	44,928	184
	Total	**45,146**	**402**
Server (Root)	**Mobility**	44,816	92
	DIS Method	44,850	126
	IDS	45,054	330
	Total	**45,272**	**548**

Table 8
Maximum ram usage of SLM-RPL in different in slm-rpl.

Node Type	Configuration	State	Max RAM (byte)
Mobile Clients	**Mobility**	**Average**	128
		Worst	142
	IDS Firewall	**Average**	14
		Worst	68
	Total	**Average**	**146**
		Worst	**210**
Static Clients	**Mobility**	**Average**	40
		Worst	42
	DIS Method	**Average**	98
		Worst	144
	IDS	**Average**	82
		Worst	95
	Total	**Average**	**220**
		Worst	**279**
Server (Root)	**Mobility**	**Average**	40
		Worst	42
	DIS Method	**Average**	98
		Worst	144
	IDS Firewall	**Average**	151
		Worst	368
	Total	**Average**	**289**
		Worst	**554**

MERPL), SLM-RPL considers mobile nodes as leaf nodes which is a logical assumption in dynamic real-world applications like healthcare, smart roads, automotive, and smart cities. Nevertheless, the performance of SLM-RPL when mobile nodes can be the parents of other nodes may decrease, which can be investigated in future studies.

4- Not choosing a proper value for the ψ parameter in the proposed IDS: as shown in the evaluation section, choosing a high value for ψ parameter decreases the TPR while, in the presence of collusion attackers, choosing a small value for this parameter can increase FPR. Therefore, choosing a proper value for the ψ parameter can be tricky, and a trade-off between FPR and TPR is needed.

8. Conclusion

In this study, a novel extension for the RPL protocol called Secured Location-Aware Mobility-enabled RPL (SLM-RPL) is proposed to manage mobility in the RPL better. SLM-RPL aims to address the weaknesses of the related mobility extensions: ineffectiveness of the mobility management procedure and not considering the security-related aspects.

From the mobility management point of view, SLM-RPL involves four building blocks:

(1) Location Propagation: Embedding geographical coordinates of static nodes in DIO messages and storing this information by the mobile receivers.

(2) Periodical Movement Detection: Periodical movement detection and calculating the distance between mobile nodes and their preferred parents, (3) Exit Prediction: Sending DIS messages by mobile nodes before getting disconnected from their preferred parents in case of a possible exit (i.e., if no proper in-range static node has been found in the parent list), and (4) Connecting to a new parent: Selecting a new proper in-range preferred parent. Using this lightweight method, mobile nodes can prepare themselves before a possible exit from their current preferred parent's range and select another preferred parent as fast as possible in case of exiting.

According to the extensive evaluations related to the mobility management part, SLM-RPL significantly reduces Packet Loss Rate (PLR) compared to other mobility management schemes, even in big, dense, or highly dynamic networks. Moreover, using a fusion formula, it has been shown that SLM-RPL is mostly the best option to be used in IoT applications, especially loss-sensitive ones. Also, SLM-RPL produces small numbers of control packets and low memory overhead.

Furthermore, unlike other mobility extensions, the security aspects of SLM-RPL have been considered. The DIS attacks are frequently used in RPL mobility extensions, and can be misused by malicious nodes to consume the resources of normal nodes. Therefore, in SLM-RPL, a probability-based method has been proposed, which is shown to be able to counter DIS attacks effectively. Also, a performable attack on SLM-RPL called False-Location-Injection (FLI) attack has been introduced, which can decrease the performance of SLM-RPL. Finally, a lightweight Intrusion Detection System (IDS) has been provided to counter FLI attack, as well as Sybil, Rank, Sinkhole, and Impersonation attacks, which all are crucial attacks to dynamic networks. The proposed IDS can also mitigate the impact of False-Reporting attacks and Collusion attacks. According to the comprehensive evaluations and comparisons with related work, the proposed IDS can efficiently counter the mentioned attacks in different scenarios with an accuracy of more than 0.99.

Finally, there could be some possible future studies, such as customizing the SLM-RPL mobility management part and evaluating the performance of SLM-RPL when mobile nodes can be the parents of other nodes, considering the velocity of the nodes in addition to the distance to the preferred parent in exit detection. Also, other abnormalities can be added to the proposed IDS to detect more attacks.

Credit author statement

All of the authors contributed to all parts of the study, Saeed Jalili: Advisor. Erfan Arvan: Student. Mahshad Koohi Habibi Dehkordi: Student.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

A Almusaylim, Z., Jhanjhi, N., Alhumam, A., 2020. Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. Sensors 20 (21), 5997.

Agiollo, A., et al., 2021. DETONAR: Detection of Routing Attacks in RPL-Based IoT, vol. 18. IEEE Transactions on Network and Service Management.

Airehrour, D., Gutierrez, J.A., Ray, S.K., 2019. SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things. Future Generat. Comput. Syst. 93, 860–876.

Almusaylim, Z.A., Alhumam, A., Jhanjhi, N., 2020. Proposing a secure RPL based internet of things routing protocol: a review. Ad Hoc Netw. 101, 102096.
Aris, A., Oktug, S.F., Yalcin, S.B.O., 2016. RPL version number attacks: in-depth study. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE.
Baghani, A.S., Rahimpour, S., Khabbazian, M., 2020. The DAO induction attack against the RPL-based internet of things. In: 2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE.
Canbalaban, E., Sen, S., 2020. A cross-layer intrusion detection system for RPL-based Internet of Things. In: International Conference on Ad-Hoc Networks and Wireless. Springer.
Cobarzan, C., Montavont, J., Noel, T., 2014. Analysis and performance evaluation of RPL under mobility. In: 2014 IEEE symposium on computers and communications (ISCC). IEEE.
Committee, L.M.S., 2003. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). IEEE Computer Society.
Dogan, C., Yilmaz, S., Sen, S., 2022. Analysis of RPL objective functions with security perspective. In: 11th International Conf. on Sensor Networks, pp. 71–80.
El Korbi, I., et al., 2012. Mobility enhanced RPL for wireless sensor networks. In: 2012 third international conference on the network of the future (NOF). IEEE.
Fotouhi, H., Moreira, D., Alves, M., 2015. mRPL: boosting mobility in the internet of things. Ad Hoc Netw. 26, 17–35.
Fotouhi, H., et al., 2017. mRPL+: a mobility management framework in RPL/6LoWPAN. Comput. Commun. 104, 34–54.
Gaddour, O., Koubâa, A., 2012. RPL in a nutshell: a survey. Comput. Network. 56 (14), 3163–3178.
Gaddour, O., et al., 2014. Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism. In: Proceedings of the 9th IEEE International Symposium on Industrial Embedded Systems. SIES, 2014. IEEE.
Ghaleb, B., et al., 2018. Addressing the DAO insider attack in RPL's Internet of Things networks. IEEE Commun. Lett. 23 (1), 68–71.
Guo, G.A., 2021. Lightweight countermeasure to DIS attack in RPL routing protocol. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC). IEEE.
Hong, K.-S., Choi, L., 2011. DAG-based multipath routing for mobile sensor networks. In: ICTC 2011. IEEE.
ISO, I., 1989. Information processing systems open systems interconnection basic reference model-part 2, 7498-2. In: Security Architecture. ISO Geneva, Switzerland.
Kamble, A., Malemath, V.S., Patil, D., 2017. Security attacks and secure routing protocols in RPL-based Internet of Things: Survey. In: 2017 International Conference on Emerging Trends & Innovation in ICT (ICEI). IEEE.
Le, A., et al., 2016. A specification-based IDS for detecting attacks on RPL-based network topology. Information 7 (2), 25.
Lee, K.C., et al., 2012. RPL under mobility. In: 2012 IEEE consumer communications and networking conference (CCNC). IEEE.
Mahbub, M., 2020. Progressive researches on IoT security: an exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics. J. Netw. Comput. Appl. 168, 102761.
Manikannan, K., Nagarajan, V., 2020. Optimized mobility management for RPL/6LoWPAN based IoT network architecture using the firefly algorithm. Microprocess. Microsyst. 77, 103193.
Medjek, F., et al., 2015. Analytical evaluation of the impacts of Sybil attacks against RPL under mobility. In: 12th International Symposium on Programming and Systems (ISPS). IEEE.
Medjek, F., et al., 2017. A trust-based intrusion detection system for mobile rpl based networks. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData). IEEE.
Montenegro, G., et al., 2007. Transmission of IPv6 Packets over IEEE 802.15. 4 Networks, vol. 4944. Internet proposed standard RFC, p. 130.
Murali, S., Jamalipour, A., 2019. A lightweight intrusion detection for sybil attack under mobile RPL in the internet of things. IEEE Internet Things J. 7 (1), 379–388.
Oliveira, A., Vazao, T., 2016. Low-power and lossy networks under mobility: a survey. Comput. Network. 107, 339–352.
Osman, M., et al., 2021. Artificial neural network model for decreased rank attack detection in RPL based on IoT networks. Int. J. Netw. Secur. 23, 496–503.
Perazzo, P., et al., 2017. An implementation and evaluation of the security features of RPL. In: International Conference on Ad-Hoc Networks and Wireless. Springer, pp. 63–76.
Prathapchandran, K., Janani, T., 2021. A trust aware security mechanism to detect sinkhole attack in RPL-based IoT environment using random forest–RFTRUST. Comput. Network. 198, 108413.
Pu, C., 2018. Mitigating DAO inconsistency attack in RPL-based low power and lossy networks. In: 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC). IEEE.
Pu, C., 2020. Sybil attack in RPL-based internet of things: analysis and defenses. IEEE Internet Things J. 7 (6), 4937–4949.
Pu, C., Choo, K.-K.R., 2022. Lightweight sybil attack detection in IoT based on Bloom filter and physical unclonable function. Comput. Secur. 113, 102541.
Raza, S., Wallgren, L., Voigt, T., 2013. SVELTE: real-time intrusion detection in the internet of things. Ad Hoc Netw. 11 (8), 2661–2674.
Sharma, M., Elmiligi, H., Gebali, F., 2021. A novel intrusion detection system for RPL-based cyber-physical systems. IEEE Can. J. Electr. Comput. Eng. 44, 246–252.
Sheibani, M., Barekatein, B., Arvan, E., 2022. A lightweight distributed detection algorithm for DDAO attack on RPL routing protocol in Internet of Things. Pervasive Mob. Comput. 80, 101525.
Simoglou, G., et al., 2021. Intrusion detection systems for RPL security: a comparative analysis. Comput. Secur. 104, 102219.
Team, G.P., 2014. Global Positioning System (Gps) Standard Positioning Service (Sps) Performance Analysis Report. GPS Product Team, Washington, DC, USA.
Thulasiraman, P., Wang, Y., 2019. A lightweight trust-based security architecture for RPL in mobile IoT networks. In: 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE.
Tsao, T., et al., 2015. *A Security Threat Analysis for the Routing Protocol for Low-Power and Lossy Networks (RPLs)*. RFC 7416 (Informational), Internet Engineering Task Force.
Verma, A., Ranga, V., 2019. ELNIDS: ensemble learning based network intrusion detection system for RPL based Internet of Things. In: 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU). IEEE.
Verma, A., Ranga, V., 2020a. Mitigation of DIS flooding attacks in RPLbased 6LoWPAN networks. Trans. Emerg. Telecommun. Technol. 31 (2) e3802.
Verma, A., Ranga, V., 2020b. Security of RPL based 6LoWPAN networks in the internet of things: a review. IEEE Sensor. J. 20 (11), 5666–5690.
Winter, T., et al., 2012. RPL: IPv6 routing protocol for low-power and lossy networks. rfc 6550, 1–157.
Yılmaz, S., Aydogan, E., Sen, S., 2021. A transfer learning approach for securing resource-constrained IoT devices. IEEE Trans. Inf. Forensics Secur. 16, 4405–4418.
Yugha, R., Chithra, S., 2020. A survey on technologies and security protocols: reference for future generation IoT. J. Netw. Comput. Appl. 169, 102763.

Erfan Arvan received an M.S. degree in computer engineering from Tarbiat Modares University, Tehran, Iran, in 2018. His main research interests are Smart Hospitals, IP-based Internet of Things, and machine learning.

Mahshad Koohi Habibi Dehkordi received an M.S. degree in computer engineering from Tarbiat Modares University, Tehran, Iran, in 2018. Her main research interests are machine learning, image processing, and Natural Language Processing (NLP), and IoT.

Saeed Jalili received his Ph.D. in Computer Science from Bradford University, Bradford, UK, in 1991. He is an associate professor in Electrical and Computer Engineering at Tarbiat Modares University, Tehran, Iran. His main research interests are security protocol verification, network security, machine learning, and software runtime verification.