Proposal and Implementation of a Security Enhancement Method using Route Hopping MTD for Mesh Networks

Yuto Ikeda
Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University
Fukuoka, Japan
e-mail: ikeda.yuto.181@s.kyushu-u.ac.jp

Hiroshi Koide
Research Institute for Information Technology, Kyushu University
Fukuoka, Japan
e-mail: koide@cc.kyushu-u.ac.jp

***Abstract*—Mesh networking has recently garnered significant attention in the Internet of Things (IoT) domain. In a mesh network, the nodes are interconnected in a mesh topology; compared to the star topology traditionally employed in many IoT systems, it offers greater fault tolerance through dynamic routing and an extended communication range. Although these advantages have led to the widespread adoption of mesh networks, the practice of forwarding packets across heterogeneous devices introduces notable security vulnerabilities. This paper presents the design and implementation of an IoT communication scheme that integrates Moving Target Defense (MTD) mechanisms—previously studied mainly in IP networks—into mesh-based IoT environments. The implemented scheme improves security by extending the conventional Ad hoc On-demand Distance Vector (AODV) protocol and applying MTD to route selection. In this method, multiple candidate paths are discovered during route exploration and one route is randomly selected for each packet when forwarding packets. The scheme mitigates man-in-the-middle and Denial-of-Service (DoS) attacks originating from a single compromised node by dynamically selecting and rotating among multiple routing paths. To evaluate performance, we implemented the proposed method in Python for Raspberry Pi and we measure and compare the processing time of the proposed scheme with that of ordinary simple Ad hoc On-demand Distance Vector (AODV) routing.**

Keywords-IoT; Mesh Network; Moving Target Defense; AODV.

I. INTRODUCTION

A. Purpose

In this paper, we implement a new security method that combines *route hopping*—a form of Moving Target Defense (MTD)—with the Ad hoc On-demand Distance Vector (AODV) routing protocol in IoT mesh networks. In conventional mesh networks, a route is used continuously once it has been discovered, which creates security concerns. This paper addresses that issue by applying a Moving Target Defense approach to improve security. Our specific contributions are summarized below:

1) Route-Hopping MTD Design: We implement the scheme that randomly assigns the packet-forwarding path per packet in an AODV-based mesh, adding some lightweight extensions.
2) Prototype Implementation and Evaluation: We implemented the Algorithm in Python on Raspberry Pi nodes with Bluetooth Low Energy (BLE). We implemented a virtual-mesh network using Raspberry Pi's BLE, and tested the Route-Hopping MTD Algorithm. The evaluation shows that security can be improved with only minimal overhead. As the first case of experimentation in a real environment, we obtained results consistent with previous simulation-based studies, thereby confirming the feasibility of the approach in practice. In particular, when conducting an experiment with five nodes and two candidate routes, the overhead was limited to only an additional 1 ms in packet forwarding time. These findings indicate that distributing traffic across multiple routes can realistically improve security without imposing significant performance overhead.

B. Motivation

In recent years, IoT devices have proliferated explosively, and by connecting a wide variety of equipment to a network, they now provide an equally diverse range of functions. These functions require a communication network, yet supporting large numbers of devices over wide areas with a conventional star topology is expected to become difficult from both cost and radio-congestion perspectives. A communication technique that addresses these issues is the mesh network. A mesh network interconnects multiple devices in a lattice-like topology and has recently been introduced into many IoT products. It is specified in numerous IoT-oriented wireless standards such as ZigBee [1], 6LoWPAN [2], Thread [3], and Matter [4]. Typical use cases span home automation scenarios, from temperature sensing and air-conditioner control to door lock, and adoption is advancing even in security-critical domains such as access control.

Moving Target Defense (MTD) [5] has attracted attention as a security technique for communication networks, including the Internet. MTD enhances security by dynamically altering system parameters such as identifiers exemplified by IP addresses or packet forwarding routes, making it harder for attackers to formulate a concrete attack strategy. This study focuses on the security of the IoT mesh network and proposes a robust security method for MTD-based mesh networks. The

proposed approach seeks to minimize security impacts even when a malicious node joins the network while remaining simple enough to run on resource-constrained embedded devices. In addition, we implemented the method using Bluetooth as a prototype to emulate real-world wireless communication and evaluate its performance to verify its practical viability.

C. Structure of this paper

The paper is structured as follows: Section II provides a background on the implementation of Route-Hopping MTD. We conclude with how MTD is usable for mesh networks. Section III introduces some related works and explains difference between the related works and this paper. Section IV then provides the Threat Model discussed in this paper. We then continue in Section V on the implementation of Route-Hopping MTD and then see the result of the metrics. Section VI provides a conclusion and gives an outlook.

II. Background

A. MTD

Moving Target defense (MTD) is a security technique that interferes with attackers by dynamically altering parameters—such as the identifiers of the resources being protected. The parameters subject to change can be broadly categorized as follows [5]:

- Identifiers used during communication (e.g., IP addresses, port numbers, MAC addresses).
- Communication paths used during packet forwarding.
- Software-execution environments (e.g., instruction sets, system-call numbers, Software Development Kits (SDKs).
- The software binaries themselves.

The primary objective of MTD is to heighten system complexity and parameter uncertainty, thereby increasing the difficulty of every stage from reconnaissance to exploitation and ultimately lowering an attacker's probability of success. By continually shifting system parameters, MTD raises the cost of reconnaissance and execution for adversaries while simultaneously reducing the likelihood that their attacks will succeed, thus strengthening overall system security.

B. Mesh Network

A mesh network is a form of network topology in which every node is interconnected in a lattice-like (mesh) structure, and it is especially prevalent in IoT deployments. This architecture is widely employed for embedded systems, e.g., ZigBee [1] and 6LoWPAN [2]. Because nodes dynamically connect to one another in a mesh, communication with distant nodes can be achieved relatively inexpensively and with minimal complexity. In a mesh network, communication occurs when a packet travels from a source node to its destination by being forwarded through several intermediate (relaying) nodes. By virtue of its dynamically routed, interwoven links, a mesh network can deliver long-range connectivity, high reliability, and excellent scalability at low cost. The routing mechanism itself is described in a later section.

C. Routing in Mesh Network

Ad hoc On-demand Distance Vector (AODV) protocol is a routing protocol widely employed for mesh networks [6]. AODV discovers packet paths by exchanging two control packets: Route Request (RREQ) and Route Reply (RREP). RREQ packets are forwarded via broadcast communication. The information included in an RREQ packet is as follows:

- Source address.
- Destination address.
- Sequence number.
- Cumulative communication cost.

RREP packets are forwarded via unicast communication. The information included in an RREP packet is as follows:

- Source address.
- Destination address.

In AODV, packet forwarding paths are discovered using the following procedure:

- The source node broadcasts an RREQ containing the destination's address to all immediate neighbors.
- Upon receiving the RREQ, each neighbor records a reverse route to the source using the header information and increments the cost metric to reflect the additional hop.
- The neighbor rebroadcasts the updated RREQ to its own neighbors.
- When an intermediate node receives an RREQ with the same sequence number it has already processed, it discards the duplicate; otherwise, it repeats the reverse-route recording and cost increment before forwarding.
- Steps 2–4 continue until the RREQ reaches the destination node.
- The destination may receive multiple RREQs; it retains only the one with the lowest cumulative cost and discards the others.
- The destination unicasts an RREP back toward the source along the reverse path stored in each intermediate node.
- When the source node receives the RREP, it caches the forward route, completing path setup so that data communication can begin.

In this way, each node—despite lacking a complete view of the entire path from source to destination—can still maintain the routing information needed to utilize the lowest-cost route. Each node can determine the next hop toward the destination node when forwarding a packet, thereby achieving shortest-path routing. Furthermore, RREP packets are delivered unicast along the routing information constructed in this procedure, and their reception signals that route discovery has completed. These mechanisms enable communication between non-adjacent nodes in a mesh network.

D. Security concerns in mesh network

Several security challenges have been identified for mesh networks such as ZigBee. Olawumi et al. [7] report concrete security concerns in ZigBee and even demonstrate proof-of-concept attacks that exploit these vulnerabilities. The same

study also highlights the risk of shared-key leakage from vulnerable devices. Because embedded systems typically face strict resource constraints, they often cannot adopt heavy-weight cryptographic mechanisms or Software-Defined Network (SDN)–based MTD techniques commonly used in general IT systems; as a result, securely storing shared keys may not always be feasible.

E. Value of implementing MTD in IoT domains

Navas et al. [8] observe that IoT devices whose parameters tend to remain static over long periods are prone to security vulnerabilities. While the authors identify Moving Target Defense as an effective countermeasure, they also note that research on applying MTD specifically to IoT systems has not yet reached full maturity.

F. Multipath AODV and Route Hopping MTD for mesh networks

Ikeda et al. [9] propose a Route-Hopping MTD scheme. Their approach enhances security by extending AODV to handle both route discovery and route selection. Specifically, multiple paths are discovered via broadcast during the discovery phase, and a route is chosen at random for each packet during forwarding, thereby improving security. Table 1 shows the results of the simulation implemented in Python. The authors implemented a simulation for the proposed method and tested its performance using the simulation. The results show that, in random number-based AODV, the route discovery time incurs only a slight overhead of about 6% respectively, compared to conventional AODV. Also, there is no overhead between simple AODV and packet id based AODV in route discovery. For packet forwarding, the performance of packet ID-based AODV remains nearly identical to that of ordinary AODV, whereas random number-based AODV introduces an additional overhead of approximately 7%. Although the evaluation was conducted only in software simulation, the results suggest that the overhead can be expected to remain sufficiently small. In this paper, we implement the proposed method and conduct a performance evaluation.

TABLE I. THE RESULTS OF SIMULATION

(unit: ms)	AODV	packet id	random number
Discovery	264	261	282
Forwarding	251	250	270

III. RELATED WORK

A. Moving Target Defence in IP Networks

Among the application domains of MTD, IP-network MTD is particularly well studied. The parameters chosen for alteration can vary, but they are generally grouped into two categories:

- Identifiers such as IP addresses and port numbers.
- Transmission-path settings such as routing tables [10].

To date, little work has explored dynamically modifying routing tables in mesh networks. A key reason is that embedded computers often cannot provide a sufficiently large and stable node population to support reliable route hopping. By contrast, the recent explosive growth of smart-home devices means that IoT mesh networks now contain enough nodes to make such path-switching increasingly practical.

B. SNR-Based multipath AODV method

Park et al. [11] propose a multipath AODV scheme that selects the optimal route according to prevailing radio conditions. Like the present work, it discovers multiple paths via AODV, but then chooses the single route with the highest communication quality. An attacker, however, could manipulate that quality—e.g., by selective jamming—to steer traffic onto a path of their choosing, after which only that "best-quality" route would keep being used. Hence, the scheme is not considered conducive to improving security.

C. Moving Target Defence for Communication Technologies in IoT Devices

Mercado-Velázquez et al. [12] propose an MTD technique that uses random communication methods in IoT devices. The proposed method distributes each device's traffic among Wi-Fi, BLE, ZigBee, and LoRa, and experiments confirm that security can be improved while limiting additional overhead such as CPU processing time to within 30 percent. However, the approach targets IoT devices that communicate directly with a server, so its applicability to the mesh networks examined in this paper is limited. Moreover, because each device must be equipped with multiple radio technologies, the method is unlikely to be practical for real-world products in terms of cost and power consumption.

D. Moving Target Defence for Communication Messages in IoT Devices

Kusumi et al. [13] propose applying MTD to communications that use the Message Queuing Telemetry Transport (MQTT) protocol. MQTT follows a publish/subscribe model in which a single publisher node that generates data and multiple subscriber nodes that receive it communicate through a processing server called a broker. Designed as a lightweight protocol running over Transmission Control Protocol / Internet Protocol (TCP/IP), MQTT is well suited to relatively long-distance IoT communications that pass through an intermediary server.

Kusumi et al. introduce an MTD technique for MQTT in which the topics—identifiers that link publishers and subscribers to specific data streams—are periodically changed. By shuffling these tokens, the method makes it difficult for a malicious attacker to track a particular topic or device. The scheme also incorporates authentication and encryption, demonstrating that MTD can effectively reinforce security in MQTT-based systems.

The key difference between that prior work and the present study lies in their respective scopes and layers of defense.

The earlier research targets IoT communications that traverse a server and proposes an application-layer security mechanism, whereas our work focuses on direct device-to-device communication within a mesh network and enhances security by modifying network-layer routing mechanisms.

E. SDN Based route mutation for MTD

Zhang et al. [14] introduce an MTD technique for wireless sensor networks which randomizes traffic between paths based on Software-defined networks. The proposed technique enhances security by forwarding requests and responses over disjoint paths, thereby increasing resilience against eavesdropping attacks. However, because it relies on Software-Defined Networking (SDN) controllers for dynamic path reconfiguration, the scheme is unsuitable for highly resource-constrained IoT devices.

F. OpenFlow Switch based MTD for sensor networks

Anajemba et al. [15] also introduce an MTD technique for sensor networks which is based on OpenFlow Switches and changes packet paths periodically. This method enhances security by pre-configuring multiple communication paths on the OpenFlow switches and periodically switching to a different path for each communication interval. This approach is designed for IP networks and relies on Software-Defined Networking (SDN), which places an excessive load on IoT devices. Therefore, it cannot be applied to mesh networks built from low-cost devices.

IV. Threat model assumed in this paper

This section defines the assumed system environment, adversary model, and security objectives, thereby clarifying what requirements the proposed *Route-Hopping MTD* must satisfy and which threats it is designed to counter.

A. System Model

The target system is a wireless mesh network composed of n IoT devices, where n ranges from ten to several hundreds. The underlying physical and link technologies are left unspecified. Also, we assume that any data exceeding a certain size is segmented into multiple packets before transmission.

B. Attacker Model

The adversary can fully compromise up to t *nodes* ($t \ll n$) inside the network. A compromised node possesses the following capabilities, while all other nodes behave correctly:

- Evaesdropping, Modifying and Dropping packets.
- Modifying RREQ/RREP packets to attract packet routes to the malicious node itself.

The adversary is *not* assumed to perform, nor is the system required to defend against, the following:

- Large-scale Radio Frequency (RF) jamming that disrupts the entire network.
- Exploitation of software vulnerabilities such as buffer overflows—attacks that fall outside the scope of this study.

C. Security Goals

The assets to be protected and the corresponding security goals are:

- Confidentiality and Integrity — Application payloads must remain undisclosed.
- Availability — The communication service should continue, keeping the Packet Delivery Ratio (PDR) as high as possible, even in the presence of compromised nodes.

and the following are not in the scope of this paper:

- Interception of packets at relay nodes and their decryption due to inadequate encryption.
- Software vulnerabilities like overflow not related to packet forwarding.

V. Design of Route-Hopping MTD in Mesh Networks

A. Overview

Previous work has paid only limited attention to MTD techniques for IoT devices, and the existing studies mainly focus on client-server communication security, not peer-to-peer style mesh network security. However, as IoT systems continue to expand, a robust security mechanism that can be applied to the mesh-network architecture will become essential, and MTD can be one promising candidate. Accordingly, this paper implements a lightweight Route-Hopping MTD scheme suitable for mesh networks suggested in [9]. We first explain the route-discovery procedure of the proposed method, and then describe two alternative strategies for selecting a path during packet forwarding.

B. Design Principles

Route-Hopping MTD is designed as the extension to the ordinal AODV, covering two main aspects: route discovery and packet forwarding. First, during route discovery, our extended AODV discovers and stores multiple candidate routes whereas ordinal AODV retains only the single shortest path. Second, when packets are transmitted, the scheme dynamically switches among these stored routes on a per-packet basis, thereby realizing route hopping. Packet forwarding methods are as follows:

- Simple packet id based shuffling.
- Packet id based random number shuffling.

The detailed method will be described later.

C. Route Discovery

Route discovery employs an enhanced version of AODV known as Multipath AODV. By extending the original protocol, this method can discover multiple routes instead of only the single lowest-cost path. Each node stores the N lowest-cost routes, according to the predetermined value of N. In our method, packet forwarding paths are discovered using the following procedure:

- The source node broadcasts an RREQ containing the destination's address to all immediate neighbors.

- Upon receiving the RREQ, each neighbor records a multiple reverse route to the source using the header information and increments the cost metric to reflect the additional hop.
- The neighbor rebroadcasts the updated RREQ to its own neighbors.
- When an intermediate node receives an RREQ with the same sequence number it has already processed, it stores up to N next-hop entries in ascending order of cost. It also updates the cost metric to record the additional hop.
- Steps 2–4 continue until the RREQ reaches the destination node.
- The destination may receive multiple RREQs; the destination node keeps the N lowest-cost next-hop entries.
- The destination broadcasts an RREP back toward the source along the reverse paths with the same manner as RREQ.
- When the source node receives the RREPs, it caches the forward routes, completing paths setup so that data communication with multiple routes can begin.

D. Route Selection and Packet Forwarding

After multiple routes have been discovered, each node has to choose one node among them when forwarding a packet. In this proposed method, the nodes determine the route based on the specific packet id. We tried two concrete approaches for selecting the packet forwarding route:

- Simple packet id-based: Given a packet identifier *pid* and the total number of stored routes *N*, the next hop is selected by computing

$$\text{index} = (pid \bmod N) + 1$$

  and the packet is forwarded along the *index*-th route in the routing table. This method is very lightweight and is very usable for resource-limited embedded systems, but the next route is easy to be guessed so the security level is lower than the random number based method described below.
- Packet id- and random number- based: Given a packet identifier pid, the total number of stored routes N, and Random Number Calculation Function

$$\text{Random}(seed)$$

  the next hop is selected by computing

$$\text{index} = (\text{Random}(pid) \bmod N) + 1$$

  and the packet is forwarded along the *index*-th route in the routing table. This method needs some calculations and might be heavier compared to the simple packet id method, but it can achieve higher security level because the next route is getting hard to guess. The impact of performing additional processing is evaluated in Section IV, Subsection C based on the simulation results.

E. Summary and Open Issues

Research on Moving Target Defense (MTD) on networks has been widely explored in the context of IP networks, focusing on altering identifiers such as IP addresses and ports or dynamically adjusting routing tables. However, these approaches generally assume resource-rich environments and are not directly applicable to IoT mesh networks. In the IoT domain, several studies have examined the application of MTD, for example by switching between multiple wireless technologies or modifying application-layer tokens in protocols such as MQTT. While these approaches demonstrate security benefits, they often rely on devices equipped with multiple radio interfaces or the presence of centralized servers, which limits their applicability in low-cost, resource-constrained mesh topologies.

Other research has extended AODV to multipath discovery, often using criteria such as signal strength to select the best route. Although effective in improving communication quality, these methods remain vulnerable to adversaries capable of manipulating perceived channel conditions, and therefore do not fully address the security problem. Similarly, SDN-based MTD techniques provide flexible route mutation, but their reliance on centralized controllers makes them unsuitable for lightweight IoT deployments.

With respect to Route-Hopping MTD specifically, previous studies have investigated its potential through simulation, showing that dynamically alternating paths can reduce the probability of an attacker consistently intercepting packets. However, to the best of our knowledge, no prior work has implemented and evaluated Route-Hopping MTD in real communication systems or on actual IoT devices. This gap highlights the lack of empirical evidence on its practicality and effectiveness in real-world mesh networks.

In summary, the existing body of work highlights two key gaps. First, most prior research remains either at the simulation stage or targets scenarios with more capable devices, leaving open the question of how MTD can be realized in practice on resource-limited IoT mesh networks. Second, while multipath discovery has been studied, only limited attention has been paid to leveraging route hopping strategies as a direct security mechanism at the network layer in mesh environments. Addressing these gaps motivates the present study.

VI. Implementation and Evaluation of the Proposed Method

A. Implementation Overview

To evaluate the routing algorithm described above, we implemented the prototype with Bluetooth and Raspberry Pi and evaluated its performance. In this section, we explain the implementation and performance evaluation results.

B. Architecture

The hardware and software used in our evaluation are as follows:

- Raspberry Pi 4 * 5 units.

- OS: Raspbian 12 Bookwarm.
- Software: Python 3.13 with Pybluez library.
- Radio: Bluetooth LE.

The software architecture is shown in Figure 1. To balance prototype development with algorithmic research, the developed software is a two-layer structure which consists of a routing algorithm layer and a Hardware Abstraction Layer (HAL). The routing algorithm layer is the core software and performs route discovery and next hop determination, while the HAL is replaceable software which is responsible for communication over real protocol stacks such as BLE and TCP.

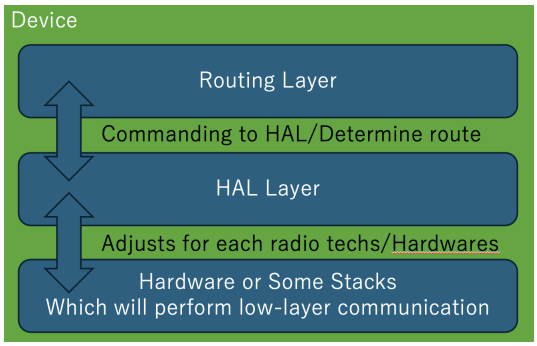

Figure 1. Route Hopping MTD Architecture.

C. Performance Evaluation

We evaluated the performance of the network and measured some metrics. Measurements were performed ten times, and the average value was calculated. The network structure is shown in Figure 2. We measured the total time of route discovery and one-packet forwarding. The results are shown in Table 2. Route discovery is done with minimal overhead compared to the simple AODV method. Packet forwarding is done with very little overhead compared to the simple AODV method.

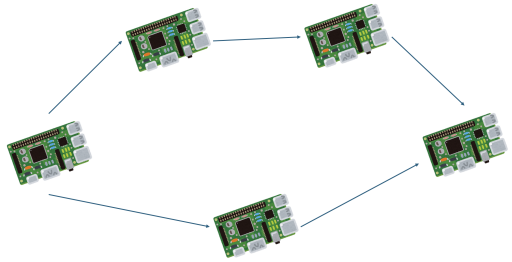

Figure 2. Network architecture for lightweight evaluation.

TABLE II. THE RESULTS OF EVALUATION

(unit: ms)	AODV	packet id	random number
Discovery	318	382	379
Forwarding	129	130	130

D. Security Evaluation

In terms of security, the system now exhibits sufficiently high resilience against threats of the kind specified in the threat model. The threat model assumed the presence of malicious nodes. Assuming there are X independent paths and malicious nodes are present on T of them, the probability that a packet traverses a malicious nodes is

$$P = T/X$$

and if the sender sends the same packet twice, the possibility that two packets both traverse malicious nodes is

$$P = T/X * (T-1)/(X-1).$$

Also, if a large amount of data is being transmitted in several packets, the possibility that all packets traverse malicious nodes is

$$P = \prod_{i=0}^{P_{\text{num}}-1} \frac{T-i}{X-i}.$$

From these equations, it is evident that the likelihood of an attacker successfully intercepting or disrupting all packets decreases rapidly as the number of available disjoint paths X increases. In other words, even if malicious nodes are distributed in the network, the probability of complete compromise becomes negligibly small once multiple independent routes are available. This stands in contrast to conventional AODV routing, where a single fixed path is repeatedly used and thus vulnerable to persistent attacks on that route. Moreover, the probabilistic distribution of packets across multiple routes creates uncertainty for adversaries. This uncertainty forces an attacker to compromise a significantly larger portion of the network in order to achieve the same level of disruption as in a static routing scheme. Consequently, the proposed scheme demonstrates clear security improvements by minimizing the success probability of attacks under realistic adversarial conditions.

E. Discussion

Based on the performance evaluation, Route-Hopping MTD is expected to be achievable without incurring significant performance overhead. The evaluation confirmed that the additional processing time incurred during packet forwarding by this method remains minimal, suggesting that the approach is applicable even in scenarios demanding higher real-time performance. Moreover, the security evaluation indicates that the approach is especially effective in enhancing security for large-scale mesh networks.

VII. CONCLUSION AND OUTLOOK

A. Conclusion

In this paper, we implemented Route-Hopping MTD and carried out a performance evaluation. The results show that the overhead remained sufficiently small. Security testing confirmed a substantial improvement in resilience. Together, these findings indicate that Route-Hopping MTD is a practical and effective security measure for IoT devices.

B. Future Works

These results demonstrate the usefulness of Route-Hopping MTD. However, we have yet to evaluate its performance in large-scale networks, and questions remain as to whether communication can be completed within practical time frames and whether memory and CPU consumption stay sufficiently low. Also, as future work, we plan to build a higher-fidelity simulation environment based on the metrics obtained in the present study and conduct more comprehensive experiments. Also, we will need to carry out a formal analytical evaluation of how memory and CPU consumption scale.

ACKNOWLEDGEMENT

This study was supported by JST K Program Grant Number JPMJKP24K3, Japan and Hitachi Systems.

REFERENCES

[1] "Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans)," *IEEE Std 802.15.4-2011 (Revision of IEEE Std 802.15.4-2006)*, pp. 1–314, 2011. DOI:10 . 1109 / IEEESTD . 2011 . 6012487.

[2] "Ieee standard for low-rate wireless networks corrigendum 1: Correction of errors preventing backward compatibility," *IEEE Std 802.15.4-2020/Cor 1-2022 (Corrigendum to IEEE Std 802.15.4-2020 as amended by IEEE Std 802.15.4z-2020, IEEE Std 802.15.4w-2020, IEEE Std 802.15.4y-2021, and IEEE Std 802.15.4aa-2022)*, pp. 1–22, 2023. DOI:10.1109/IEEESTD.2022.10014667.

[3] *Thread specification, revision 1.3.0*, Available: https://www.threadgroup.org/ThreadSpec, Thread Group, Inc., 2023.

[4] *Matter 1.2 specification*, Available: https : / / csa - iot . org / all-solutions/matter/, Connectivity Standards Alliance (CSA), 2023.

[5] Y. W. X. Zhou Y. Lu and X. Yan, "Overview on moving target network defense," *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*, pp. 821–827, 2018.

[6] Z. Sun, X.-G. Zhang, D. Ruan, H. Li, and X. Pang, "A routing protocol based on flooding and aodv in the zigbee network," in *2009 International Workshop on Intelligent Systems and Applications*, 2009, pp. 1–4. DOI:10 . 1109 / IWISA . 2009 . 5072672.

[7] O. Olawumi, K. Haataja, M. Asikainen, N. Vidgren, and P. Toivanen, "Three practical attacks against zigbee security: Attack scenario definitions, practical experiments, countermeasures, and lessons learned," in *2014 14th International Conference on Hybrid Intelligent Systems*, 2014, pp. 199–206. DOI:10.1109/HIS.2014.7086198.

[8] R. E. Navas, F. Cuppens, N. Boulahia Cuppens, L. Toutain, and G. Z. Papadopoulos, "Mtd, where art thou? a systematic review of moving target defense techniques for iot," *IEEE Internet of Things Journal*, vol. 8, no. 10, pp. 7818–7832, 2021. DOI:10.1109/JIOT.2020.3040358.

[9] Y. Ikeda and H. Koide, *Implementing route-hopping mtd for iot mesh networks*, Japanese, Mar. 2025. [Online]. Available: https://ipsj.ixsq.nii.ac.jp/api/records/2001195.

[10] J. Narantuya *et al.*, "Sdn-based ip shuffling moving target defense with multiple sdn controllers," in *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – Supplemental Volume (DSN-S)*, 2019, pp. 15–16. DOI:10.1109/DSN-S.2019.00013.

[11] J. Park, S. Moh, and I. Chung, "A multipath aodv routing protocol in mobile ad hoc networks with sinr-based route selection," in *2008 IEEE International Symposium on Wireless Communication Systems*, 2008, pp. 682–686. DOI:10.1109/ISWCS.2008.4726143.

[12] A. A. Mercado-Velázquez, P. J. Escamilla-Ambrosio, and F. Ortiz-Rodríguez, "A moving target defense strategy for internet of things cybersecurity," *IEEE Access*, vol. 9, pp. 118 406–118 418, 2021. DOI:10.1109/ACCESS.2021.3107403.

[13] K. Kusumi and H. Koide, "Mqtt-mtd: Integrating moving target defense into mqtt protocol as an alternative to tls," in *2024 7th International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2024, pp. 1–8. DOI:10.1109/CommNet63022.2024.10793300.

[14] B. Zhang and L. Han, "Dynamic random route mutation mechanism for moving target defense in sdn," Jun. 2021, pp. 536–541. DOI:10.1109/ISCIPT53667.2021.00114.

[15] J. Anajemba *et al.*, "Dsphr: A dynamic sdn-based port hopping routing technique for mitigating sd-wsn attacks," Apr. 2024. DOI:10.1007/s11277-024-10979-7.