Security-Oriented Load Distribution in MQTTv5: A Token Bucket and Trust Evaluation Approach

Mattia Giovanni Spina*†, Graziano Rizzo*, Floriano De Rango*†
* DIMES, University of Calabria, Via P. Bucci, 87036 Rende (CS), Italy
† CNIT - National Inter-University Consortium for Telecommunications, Viale G.P. Usberti 181/A, 43124, Parma (PR), Italy
{mattiagiovanni.spina, graziano.rizzo, derango}@dimes.unical.it,

***Abstract*—The advent of fifth-generation (5G) networks has enabled the rise of the *massive IoT* (mIoT) paradigm, characterized by large-scale, heterogeneous, and interconnected devices leading to breakthrough applications like telemedicine, AR/VR, and autonomous systems. In this context, the security of communication protocols becomes critical. Among these, MQTT is a widely adopted protocol for IoT communications. Its latest specification, MQTTv5, introduces the *shared subscription* feature to improve scalability and reduce message overhead on constrained devices. However, prior work by the same authors revealed a critical vulnerability in this mechanism, allowing attackers to induce starvation of legitimate subscribers. An initial mitigation was proposed, based on limiting the number of subscribers per group using fixed statistical thresholds. While effective, this approach compromises scalability by potentially excluding legitimate nodes. To address this, we propose a novel *adaptive, trust-based token bucket* mechanism that regulates message delivery without restricting group size. By dynamically adjusting delivery rates based on subscriber trust, the system mitigates attacks while preserving scalability. Experimental results demonstrate the superior effectiveness and efficiency of the proposed solution compared to the previous approach.**

***Index Terms*—MQTT, Shared Subscription, IoT Network Security**

I. Introduction

The Internet of Things (IoT) growth is closely lined to *wireless communication*, essential for real-time data processing in sectors like healthcare, transport, and smart cities. As IoT scales into massive IoT (mIoT), connecting billions of constrained devices, current 5G networks face scalability and latency issues. To address these, the future 6G – by natively integrating Artificial Intelligence (AI) into the network – has emerged as a key driving force. to ensure intelligent, ultra-reliable, low-latency connectivity for mIoT scenarios [1], [2]. As network architectures evolve, the application-layer protocols that underpin IoT services must evolve accordingly. In this context, the Message Queuing Telemetry Transport (MQTT) protocol has gained widespread adoption due to its lightweight nature and suitability for constrained devices and unreliable wireless links. The latest version, MQTTv5 [3], introduces several enhancements aimed at improving the scalability, flexibility, and robustness of MQTT-based deployments. Among these enhancements, the *shared subscription* feature represents a key feature to support large-scale wireless IoT scenarios. It enables multiple subscribers to share the load of processing messages published to a common topic, effectively facilitating horizontal scalability and better resource utilization. Despite the benefits introduced by the shared subscription mechanism, prior work by the author in [4] revealed a critical vulnerability that allows malicious users to exploit this feature to perform a *Distributed Denial of Service (DDoS) Starvation attack*. This attack undermines the intended scalability and efficiency gains by selectively overwhelming specific subscribers, effectively nullifying the load-balancing benefits of shared subscriptions. The previously proposed mitigation (by the same authors of this paper) in [4], based on a fixed subscriber limit per shared group, showed limitations in terms of scalability. To overcome these issues, in this paper, an adaptive and trust-based mechanism is introduced, where subscribers are grouped according to trust levels and their behavior is continuously assessed through a *reward-penalty model*. Additionally, a *Token Bucket Message Scheduling* strategy regulates message delivery across sub-groups, prioritizing high-trust subscribers. The proposed solution enables runtime, behavior-aware mitigation without imposing static constraints on group composition. Through an extensive experimental campaign, the proposed dynamic mitigation is compared against the previous static one, demonstrating the benefits of an adaptive approach, improving the scalability and the effectiveness of the system. The remainder of the paper is structured as follows: Section II reviews the related work and highlights the main contributions. Section III defines the problem formulation. Section IV presents the proposed mitigation strategy. Section V details the performance evaluation and discusses the results. Finally, Section VI concludes the paper.

II. Related works

DoS and DDoS attacks are major threats in IoT due to device heterogeneity, limited resources, and the lack of lightweight security mechanisms [5]. This has driven growing interest in mitigation strategies, especially for MQTT, a widely used IoT application protocol. In this direction, [6], [7] designed strategies to counteract volumetric DoS/DDoS in MQTTv3 deployments, resorting to lightweight Elliptic Curve cryptography. The solutions are based on organizing topics in security-related priorities, each with pre-defined guaranteed security degrees. Nonetheless, these works are focused on MQTTv3.1 and neglect advanced features of MQTTv5, like

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

shared subscription, that could natively help and enable such a type of topic organization by means of shared groups. In addition, advanced DoS/DDoS types, like starvation, could still be successfully executed, imposing indefinite delay in the message delivery process. Current advancements on future programmable networks – with SDN, NFV, and PDP paradigms – have been also exploited to support IoT network security. Following this trend, [8] exploits the holistic view of the SDN controller over the network packets to perform DDoS attacks in an IoT environment by analysing the IP packets' session counter and payload. The centralized and holistic nature of the SDN controller over the network is commonly combined with *Artificial Intelligence* (AI) to empower the network with an automatic threat detection mechanism. In [9], [10], the SDN controller is enhanced with Machine/Deep Learning (ML/DL) algorithms to counteract application layer DDoS. The authors of [11] resort to a complex Deep Reinforcement Learning (DRL) to detect attacks in IoT environments. Nevertheless, adopting the SDN architecture broadens the attack surface, exposing the network to more sophisticated threats. The centralized nature of the SDN controller exacerbates this issue, as it represents a single point of failure. The situation becomes even more critical when complex AI algorithms run on the controller, as they require significant computational resources, increasing the risk of performance degradation or even system failure under attack. In IoT-constrained environments, a protocol-aware solution is preferable,relying instead on lightweight and protocol-tailored mechanisms.

Main Contribution

Building on the research gap on MQTTv5 shared subscription security and given the importance of MQTT as an application-layer IoT protocol, the present paper extends the authors' previous work [4], which identified a vulnerability in the *shared subscription* mechanism providing an initial mitigation design.This paper aims to highlight the limitations of the prior mitigation design presented in [4]. Specifically, the mentioned mitigation was based on a fixed parameter, namely M, used to limit the number of allowed subscribers in a shared group. Despite reducing the impact of the DDoS starvation attack, this static and resource-agnostic technique potentially lead to exclude legitimate subscribers and, therefore, affect the scalability and performance of the deployment. To overcome this issue, in this work a dynamic and adaptive trust-based mitigation is designed. Exploiting the shared subscription approach, a shared group is further organized into three subgroups based on trust ranges: *low*, *mid*, and *high*-trust. The broker constantly assess the subscribers' behaviour, applying a trust punishment-reward mechanism. To mitigate the impact of the malicious subscribers, a *Token Bucket Message Scheduling* is introduced to regulate the amount of messages that can be, in a certain time, delivered to each of the considered sub-groups. This solution prevent from imposing a limit on the number of allowed subscriber for a shared group by introducing a runtime behavioural-based subscriber trust assessment and containing malicious subscribers through an adaptive mitigation.

III. Problem Formulation

MQTTv5 introduces *shared subscription* to improve scalability by delivering each message to only one subscriber in the group, reducing traffic and easing load on constrained devices. To select the node in a group, standard balancing algorithms such as Round-Robin and Random are used. However, as demonstrated in [4], [12], default selection strategies (e.g., round-robin, random) expose a critical uncovered vulnerability: attackers can exploit them to starve legacy subscribers, leading to DDoS-like effects within the group. In the following, the vulnerability identified by the authors in [4], along with the corresponding mitigation, is first briefly recalled. This paper then analyzes the limitations of that initial solution and introduces an enhanced mechanism to address them, further strengthening the security of the shared subscription feature. The proposed improvement is detailed in Section IV.

A. Shared Subscription Vulnerability and Attack Scenario

In a shared subscription with N legacy subscribers, the expected time interval between two consecutive messages for a generic legitimate subscriber is $T_i = r \cdot N$, where r is the message publish rate. This holds for both round-robin and random forwarding strategies, as previously demonstrated in prior works of the same authors [4], [12]. When K malicious subscribers join the group, the total number of subscribers becomes $N + K$, increasing the expected inter-arrival time to $T_i = r \cdot (N + K)$. In this scenario, the uncontrolled addition of malicious subscribers to the shared group – causing K to grow without bounds – results in indefinite starvation of the legitimate N subscribers. As a consequence, legitimate subscribers are unable to properly receive messages, effectively constituting a DDoS starvation attack.

B. Baseline Solution: Fixed Allowed Subscriber Number

The previous mitigation presented in [4] is hereafter described. To maintain acceptable delay in shared subscription groups, the expected time interval between two messages received by a legitimate subscriber must satisfy $T_e = T_i + T_s + \delta_t \leq T_m$, where T_s and δ_t represent application and network delays, respectively. Given $T_i = r \cdot (N + K)$, where N and K are the numbers of legacy and malicious subscribers respectively, the total number of subscribers $M = N + K$, allowed to access the shared subscription group, must satisfy:

$$M \leq \frac{T_{max} - T_s - \delta_t}{r} \tag{1}$$

This bound enables proactive mitigation by limiting the number of subscribers in a group based on T_m, thus constraining the impact of malicious nodes on message latency.

C. Threat Model

The following threat model assumptions have been considered for the design and validation of the mitigation strategy:

- The MQTT broker is assumed to be a fully trusted entity. It behaves according to the protocol specification and it is not compromised.

- External attackers are defined as entities that gain access to the MQTT system without authorization. Due to the lack of secure communication or weak authentication policies, they may connect as clients and join shared subscription groups.
- Internal attackers are legitimate clients that initially exhibit benign behavior but later adopt adversarial strategies while still complying with the MQTT protocol.
- Malicious entities follow all MQTT specifications, including the rules for joining shared subscription groups. As such, in the absence of profiling mechanisms, they are indistinguishable from legitimate nodes during recipient selection.
- All entities of the system, including attackers, conform to the MQTT standard. Malicious behavior does not rely on protocol violations, but rather on strategic exploitation of protocol-compliant interactions, such as unfair participation in recipient selection within shared groups.

IV. Dynamic and adaptive mitigation

This section presents the proposed dynamic and adaptive mitigation strategy, which is structured around three key components: (i) *Trust Monitoring and Assessment*, (ii) *Trust-based Shared Group Partitioning*, and (iii) *Token Bucket Message Delivery Regulation*.

A. Trust Monitoring and Assessment through Statistical z-score Profiling

In order to monitor and asses the behaviour of the subscribers in a shared group, a trustworthy technique is introduced. Let G be a shared group. Each subscriber $s_i \in G$ communicates to the broker the value of TM_{s_i}, as described in Section III-B. Malicious subscribers that join the shared group can deliberately communicate crafted values of TM in order to degrade the effectiveness of the mitigation proposed in [4], leading to the computation of a greater value of M – despite limiting – allowing an increasing number of malicious subscribers. The security of this subscriber-broker communication against *external attacks* has been designed and assessed in a prior work of the same author in [12], using an AEAD algorithm to secure and authenticate the communication of the TM values. However, the proposed security measure falls short when it comes to *internal attacks*, subscribers that are allowed in the shared group and that – for instance, after being hijacked – start to misbehave. To account for this issue and to provide comprehensive security, a *z-score anomaly detection* technique is introduced here. Let $s_i \in G$ be a generic subscriber of the shared group G. Let t' denote the time instant at which the trustworthiness of the node s_i is evaluated. $H^{t<t'}$ denotes the set of historical records of the TM_{s_i} values registered till $t < t'$. Let $TM_{s_i}^{t'}$ be the value communicated by the subscriber to the broker at time t'. The broker computes the *z-score* using the following formula:

$$z_{s_i}^{t'} = \frac{TM_{s_i}^{t'} - \mu_{H^{t<t'}}}{\sigma_{H^{t<t'}}} \tag{2}$$

Where $\mu_{H^{t<t'}}$ and $\sigma_{H^{t<t'}}$ represents the historical mean and the standard deviation computed on H at time t'. The computed *z-score* guides the trust update process for a subscriber of a shared group. Specifically, let $\omega_{s_i}(t)$ denote the trust value associated with the subscriber $s_i \in G$. The value of $z_{s_i}^{t'}$ provides a standardized indication of how anomalous the observed metric $TM_{s_i}^{t'}$ is, relative to the historical behaviour. A significantly large absolute value of $z_{s_i}^{t'}$ indicates a suspicious event, as it reflects that the observed $TM_{s_i}^{t'}$ value deviates substantially – i.e., by several standard deviations – from the historical mean. Such a deviation suggests anomalous behaviour that is statistically unlikely under normal conditions. Based on these considerations, a *punishment/reward* mechanism is introduced, in which the trust value is adjusted according to the number of standard deviations from the mean, i.e. the *z-score* defined in Eq.2. Formally, let $\Delta_{s_i}^{t'}$ be the trust adjustment based on the computation made in Eq.2. The trust value of the node s_i at time $t' > t$ is performed as follows:

$$\omega_{s_i}(t') := \omega_{s_i}(t) + \Delta_{s_i}^{t'} \tag{3}$$

where

$$\Delta_{s_i}^{t'} = \begin{cases} \alpha \cdot \left(1 - \dfrac{|z_{s_i}^{t'}|}{T}\right), & \text{if } |z_{s_i}^{t'}| < T \quad \text{(reward)} \\ 0, & \text{if } |z_{s_i}^{t'}| = T \quad \text{(no adjustment)} \\ -\beta \cdot \left(\dfrac{|z_{s_i}^{t'}|}{T} - 1\right), & \text{if } |z_{s_i}^{t'}| > T \quad \text{(punishment)} \end{cases} \tag{4}$$

The parameter T in Eq.4 represents the *anomaly threshold* defined for the *z-score* in order to determine whether the current value of $TM_{s_i}^{t'}$ is an outlier (i.e. an anomaly) or not. Commonly, following the *z-score* theory, when the current sample differs by $T = 2$ standard deviations, it can be considered an outlier. In Eq. 4 $\alpha, \beta \in (0, 1)$ regulates the entity of the trust variation (reward/punishment) such that:

- $\alpha > \beta$: trust updates favor reward over punishment, resulting in faster recovery and more conservative penalization.
- $\alpha = \beta$: reward and punishment are applied symmetrically.
- $\alpha < \beta$: trust increases more slowly, while penalties are applied more aggressively.

Relying on the statistical analysis performed through the *z-score* theory, the trust value of the involved subscribers is constantly monitored and assessed. The trust-based mechanism just described is not only used to detect anomalies due to *internal* malicious entities, but it also allows the design of a mitigation strategy that can confine and limit them. The proposed mitigation is described in the following subsection, and it is based on the combination of a trust-based shared group G partitioning combined with a *Token Bucket Message Delivery Regulation* technique.

B. Trust-based Shared Group Partitioning

Let G be a shared group, and let s_i be a generic subscriber in G. The group G is partitioned into three disjoint sub-groups

based on a trustworthiness criterion: G_{low}, G_{medium}, G_{high}. This partitioning of the shared group G allows the segregation and management of each subscriber $s_i \in G$ in one of the sub-groups on the basis of the trust-level $\omega_{s_i}(t)$ assumed by the node s_i at a certain time t. Given a value of $\omega_{s_i}(t) \in [0,1]$ at time t and let $\theta_1, \theta_2 \in (0,1)$ two threshold parameters such that $\theta_1 < \theta_2$. The subscriber s_i will be migrated in a G_{low}, G_{medium}, G_{high} as follows:

$$\begin{cases} s_i \in G_{\text{low}} & \text{if } \omega_{s_i}(t) \in [0, \theta_1] \\ s_i \in G_{\text{medium}} & \text{if } \omega_{s_i}(t) \in (\theta_1, \theta_2] \\ s_i \in G_{\text{high}} & \text{if } \omega_{s_i}(t) \in (\theta_2, 1] \end{cases} \tag{5}$$

This partitioning manages subscribers based on trust levels, enabling finer control over group dynamics and message dissemination. Real-time monitoring allows dynamic promotion or demotion across trust-based sub-groups using the trust score ω. Unlike [4], this strategy limits message consumption by untrusted subscribers without capping the group size. To this end, a *Token Bucket Message Delivery Regulation* mechanism is introduced, which allocates a specific number of tokens – representing the maximum number of messages that can be consumed – to each sub-group G_{low}, G_{medium}, G_{high}. The allocation is performed in such a way that sub-groups with lower trust levels receive fewer tokens, thereby limiting the message consumption rate of potentially malicious or untrusted subscribers within a given time interval. The detailed design and operational principles of this regulation mechanism are provided in the following section.

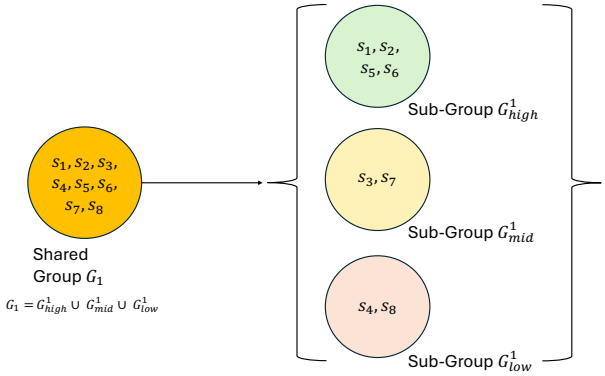

Fig. 1: Graphical Illustration of Trust-based organized shared groups.

C. Token Bucket Message Delivery Regulation

To regulate the amount of messages consumed by low-trust subscribers in G, a message delivery regulation mechanism based on a *Token Bucket scheme* is introduced and integrated with the trust mechanism previously described. Let $B(G)$ denote the bucket associated to G. In this setting, the tokens of the bucket represent the message arriving at the broker in a certain time span τ and that need to be delivered to the subscriber of the shared group G. Due to network dynamics, therefore, $B(G)$ cannot be considered as a fixed value, but its size needs to evolve keeping pace with the rate of the packets to be delivered. Assuming a message arrival rate r_G (pkt/s), estimated based on the historical behavior of the system, the minimum required tokens to ensure the delivery of all the packets arriving in a time interval τ can be defined as:

$$B(G) = r_G \cdot \tau \tag{6}$$

To ensure an optimal distribution of the $B(G)$ tokens that should be assigned to the nodes of the group, it is necessary to take into account the actual TM demand of each node in group G. When the broker needs to distribute the token each τ seconds, it can estimate, based on the TM values received by the subscribers of the group, the number of tokens that should be redistributed to each subscriber, proportionally to its TM requirement. In this context, two distinct cases can be distinguished: when $TM_{s_i} > \tau$ and when $TM_{s_i} \leq \tau$. In the first case, the subscriber s_i requires to be served less frequently than the token refresh interval τ, and therefore it is not strictly necessary to allocate a token to such a node at every distribution cycle. To determine whether a token should be granted to a generic node s_i with $TM_{s_i} > \tau$, the broker evaluates the time elapsed since the last message delivery to s_i, denoted as tl_{s_i}, and computes the residual time before reaching the maximum tolerated delay of the node as $tr_{s_i} = TM_{s_i} - tl_{s_i}$. The value of tr_{s_i} is then compared with the token redistribution interval τ and if $tr_{s_i} \leq \tau$, it means that subscriber s_i will reach its maximum tolerated delay TM_{s_i} within the current window τ, and therefore a token must be assigned to the node s_i to ensure the message delivery within the expected TM_{s_i}. The second case, where $TM_{s_i} \leq \tau$, implies that the subscriber must be served at least once within every token redistribution window, since its TM constraint will certainly expire during this interval. To determine the actual number of tokens to be allocated to subscribers in this category, it is necessary to evaluate how many times the TM_{s_i} of a subscriber s_i expires within a single redistribution window τ, because the correct amount of tokens should be distributed to it. Thus, the token demand of a subscriber s_i in this case can be computed as:

$$nT(s_i) = \left\lceil \frac{\tau}{TM_{s_i}} \right\rceil \tag{7}$$

where $nT(s_i)$ represents the number of tokens to be assigned to s_i in order to satisfy its message delivery requirements during the interval τ. Although it is possible to estimate the number of tokens required to meet the service needs of the nodes within their respective TM constraint, such estimation must be weighted according to the level of trust assigned to each node. Given $B(G)$ tokens to distribute over the subscriber in G, the proposed *Token Bucket Message Delivery Regulation* exploit the trust-based partitioning of G described in Section IV-B to implement a trust-aware delivery strategy. This approach prioritizes high-and-medium trust subscribers in G_{high}, G_{medium} while reducing the amount of tokens granted for the subscriber confined in G_{low}. This means that the tokens

in $B(G)$ are primarily allocated to serve the subscribers in G_{high}, G_{medium}; only residual tokens, if any, are then used to serve the subscribers in G_{low}. Let $\lambda_{low}, \lambda_{medium}, \lambda_{high} \in (0, 1)$ be three percentage values. Given $B(G)$ it is possible to define $B_{low}, B_{medium}, B_{high}$ as follows:

$$\begin{aligned} B_{low} &= \lambda_{low} \times B(G) \\ B_{medium} &= \lambda_{medium} \times B(G) \\ B_{high} &= \lambda_{high} \times B(G), \text{ with} \\ \lambda_{low} &< \lambda_{medium} < \lambda_{high}, \text{ and} \\ B(G) &= B_{low} + B_{medium} + B_{high} \end{aligned} \tag{8}$$

Based on Eq. 8, the allocation guarantees that $B_{\text{low}} < B_{\text{medium}} < B_{\text{high}}$. As a result, subscribers in the low-trust group – potentially exhibiting suspicious behavior –are automatically assigned a reduced number of messages. This ensures that legitimate subscribers retain access to the messages intended for them. Moreover, it is essential to ensure that each group actually holds a sufficient number of tokens to correctly deliver messages according to the proportions and rules previously described. For this reason, in addition to the time-based token refresh mechanism (based on τ), the proposed approach also triggers a token refresh whenever the bucket associated with any of the three groups becomes empty as well as a subscriber gets promoted/demoted to higher/lower trust-based sub-groups.

V. Performance Evaluation

This section delves in the description of the experimental setup and performance evaluation of the proposal.

A. Experimental Setup

TABLE I: Simulation parameters.

Parameter	Value
#Legitimate subscribers	100, 1000 and 10000
Scenarios	S_1, S_2, S_3
Malicious additional Subs.	10%, 20%, 30% and 50% of initial number
α, β	(0.3, 0.7)
$\lambda_{low}, \lambda_{mid}, \lambda_{high}$	0.1, 0.35, 0.55
θ_1, θ_2	0.33, 0.66

The attack and mitigation were simulated using the Mochi MQTT broker across three scenarios: S1 (100 nodes), S2 (1000), and S3 (10,000, representing M-IoT). Each scenario included 10% up to 50% of malicious nodes, assessing the impact on average Data Extraction Rate (DER) and mean inter-arrival time at legitimate nodes. Three configurations were tested: no security (Random *R* and Round Robin *RR*), the previous mitigation approach (**M**) [12], and the proposed Token Bucket-based mechanism (**TB**). The main simulation parameters, including the trust thresholds (θ_1 and θ_2) and the weighting factors used in Eq. 8 and Eq. 4, are summarized in Table I.

B. Trust Evaluation Results

The trust adjustment model relies on a scoring function with two tunable parameters, α and β, representing the positive and negative impacts of node behavior on their trust level. As these parameters affect the sensitivity and robustness of detection system, several configurations were tested to identify the most effective values for the final simulations. Each was evaluated through simulations with both legitimate and malicious nodes, updating trust based on reported TM values. Nodes were flagged as malicious when their trust dropped below 0.33 and they entered the G_{low} group. Effectiveness was measured using TPR, FPR, TNR, FNR, and accuracy, with results shown in Tab.II. Based on the results of this analysis, the final values of α and β used for the simulations have been set to 0.3 and 0.7, respectively. This parameter configuration yielded the best performance, ensuring higher detection precision and a reduced number of false positives, thereby minimizing the impact on legitimate nodes within the system.

TABLE II: Performance of the trust evaluation mechanism under different α and β configurations

α	β	**TPR**	**FPR**	**TNR**	**FNR**	**Accuracy**
0.3	0.7	0.933	0.01	0.99	0.067	0.978
0.7	0.3	0.100	0.02	0.98	0.9	0.782
0.5	0.5	0.533	0.015	0.985	0.467	0.884

C. Results

In terms of average inter-message delay – in Fig.2 – for legacy nodes, the proposal consistently achieves the lowest latency, with values ranging from 31.37 to 32.4 seconds at $N = 100$, 423.5 to 428.61 at $N = 1000$, and remaining scalable even at $N = 10000$. In contrast, ***RR*** and ***R*** suffer severe degradation under increasing attack intensity, reaching delays over 15000 and 13000 seconds respectively at $N = 10000$. The ***M*** approach performs better but remains less efficient, showing delays around 99, 990, and 9900 seconds for the respective network sizes. Regarding Data Extraction Rate (DER) shown in Fig.3, the *Token-based* method maintains over 94% under high attack rates and exceeds 98% under low-intensity attacks across all scales. For instance, DER at $N = 100$ ranges from 98.55% to 94.17%, and at $N = 10000$ from 99.77% to 95.39%. While ***M*** sustains DER near 94%, ***RR*** and ***R*** fall drastically to around 67% under heavy attacks. Additionally, the proposed approach admits 4% to 42% more nodes (with a variability of around $\pm 2\%$) than the previous fixed-capacity method ***M***. Unlike the former strategy limited to M subscribers, the new design allows broader inclusion by admitting all candidates and detecting adversarial behavior post-admission, preserving security and system integrity. The *Token-based* strategy stands out as the most scalable and resilient, ensuring low delay and high DER even under attack. It consistently meets delivery deadlines (TM) for all legitimate nodes, regardless of network size or attack intensity. Minor performance variations are mainly due to differences in malicious node detection speed. Nevertheless, one of the most

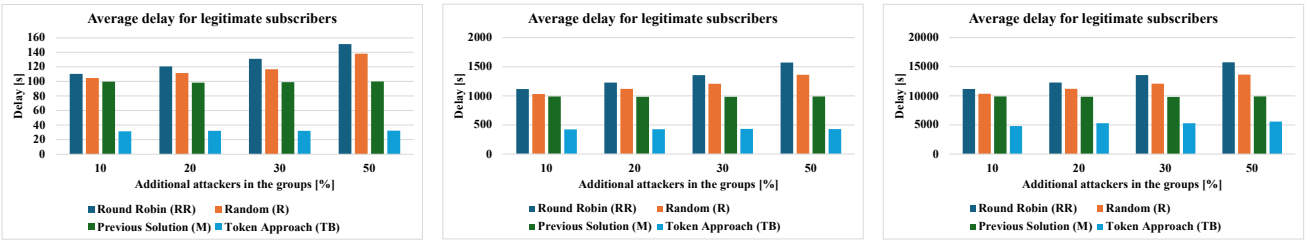

Fig. 2: Average inter-arrival time of messages with: (a) 100 nodes, (b) 1000 nodes and (c) 10000 legitimate nodes per group.

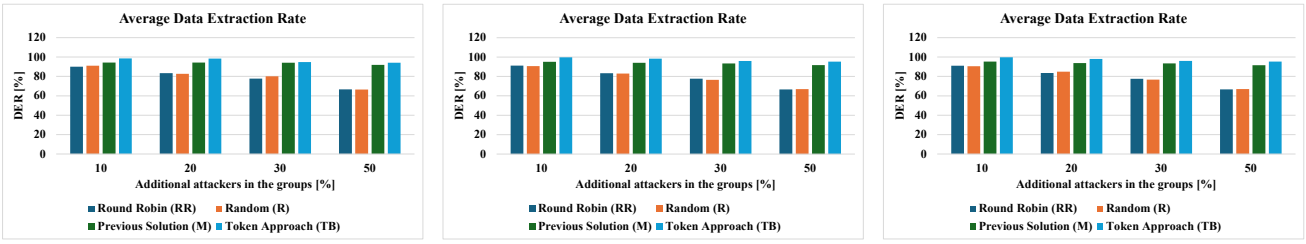

Fig. 3: Average Data Extraction Rate with: (a) 100 nodes, (b) 1000 nodes and (c) 10000 legitimate nodes per group.

important aspects is that the proposed mitigation consistently achieves strong detection performance, with high true positive rates (TPR), around 94% — indicating effective identification and segregation of attackers — and low false negative rates (FNR), around 6%, minimizing undetected threats, with an overall accuracy in the detection of around 98%. This confirms not only the efficiency of the solution under normal operating conditions but also its resilience in adversarial environments, thereby ensuring the overall reliability of the system.

VI. Conclusions

Building upon prior work by the same authors [4], [12], the security of the *shared subscription* feature introduced in MQTTv5 is further examined. A vulnerability has been identified whereby a DDoS starvation attack can be carried out by injecting a large number of fake subscribers into a shared group, potentially preventing legitimate subscribers from receiving messages. An initial mitigation based on capping group size reduces the impact of the attack but limits scalability. To overcome this, an adaptive trust-based mechanism is proposed in this work, enabling the broker to profile subscriber behavior without imposing hard limits on group size. Subscribers are clustered into trust-based subgroups, and a token-bucket policy limits message delivery to low-trust entities. Evaluation shows the approach effectively mitigates attacks while enhancing scalability and a greater number of legitimate subscribers to benefit from the shared subscription feature.

Acknowledgements

This work was partially supported by project SERICS (PE000 00014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

[1] F. Guo *et al.*, "Enabling Massive IoT Toward 6G: A Comprehensive Survey," *IEEE IoT J.*, vol. 8, no. 15, pp. 11 891–11 915, Mar. 2021.

[2] T. Kato *et al.*, "Challenges of CPS/IoT Network Architecture in 6G Era," *IEEE Access*, vol. 12, pp. 62 804–62 817, Apr. 2024.

[3] A. Banks *et al.*, *MQTT version 5.0*, OASIS Standard, 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html

[4] G. Rizzo *et al.*, "Securing Shared Subscriptions in MQTTv5 for IoT Networks: Vulnerability Analysis and Mitigation," in *2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, pp. 21–23.

[5] P. Kumari and A. K. Jain, "A comprehensive study of DDoS attacks over IoT network and their countermeasures," *Computers & Security*, vol. 127, p. 103096, Apr. 2023.

[6] F. De Rango *et al.*, "Energy-aware dynamic Internet of Things security system based on Elliptic Curve Cryptography and Message Queue Telemetry Transport protocol for mitigating Replay attacks," *Pervasive and Mobile Computing*, vol. 61, p. 101105, 2020.

[7] ——, "Mitigating dos attacks in iot edge layer to preserve qos topics and nodes' energy," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020.

[8] J. Bhayo *et al.*, "A Time-Efficient Approach Toward DDoS Attack Detection in IoT Network Using SDN," *IEEE IoT J.*, vol. 9, no. 5, pp. 3612–3630, Jul. 2021.

[9] D. Mohammed Sharif and H. Beitollahi, "Detection of application-layer DDoS attacks using machine learning and genetic algorithms," *Computers & Security*, vol. 135, p. 103511, Dec. 2023.

[10] W. I. Khedr *et al.*, "FMDADM: A Multi-Layer DDoS Attack Detection and Mitigation Framework Using Machine Learning for Stateful SDN-Based IoT Networks," *IEEE Access*, vol. 11, Mar. 2023.

[11] Zabeehullah *et al.*, "DQQS: Deep Reinforcement Learning-Based Technique for Enhancing Security and Performance in SDN-IoT Environments," *IEEE Access*, vol. 12, pp. 60 568–60 587, Apr. 2024.

[12] G. Rizzo *et al.*, "Improving IoT System Resilience through Secure MQTTv5 Shared Subscriptions," in *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*. IEEE, pp. 10–13.