Lightweight Distributed Sybil Detection and Whitewashing Mitigation in MANETs Using Mobility Correlation

Samir Selmane, Abderrahmane Baadache, and Fouzi Semchedine

***Abstract*—Characteristics of Mobile Ad Hoc Networks (MANETs), such as decentralized architecture, shared communication medium, and node mobility, make them vulnerable to a variety of security threats. Sybil attack is one of the most challenging threats in which a malicious node pretends to be multiple distinct nodes to gain disproportionate resources and disrupt network functionalities. Several traditional solutions have been proposed in the literature, but they are either extensive in resource consumption or inadequate for the MANET context. This paper introduces a distributed lightweight method that exploits Sybil identities' mobility correlation to detect the attack. It disseminates detection results using a lightweight hashing function. In addition, it uses a probation period to mitigate whitewashing attempts. Simulations using the INET framework under the OMNeT++ simulator, and an independent cross-validation under NS-3, reveal high detection accuracy up to 99.79% with a false positive rate of 0%. Our method can operate as a standalone solution or be integrated into existing routing protocols or IDSs.**

***Index Terms*—MANET, Sybil attack, lightweight detection, neighbor correlation, secure dissemination, whitewashing mitigation.**

I. Introduction

MANETs have emerged as a versatile and dynamic communication paradigm that offers transparent connectivity and data exchange among mobile devices without relying on centralized infrastructure. These autonomous networks find applications in various domains, such as disaster rescue, tactical military operations, vehicular communications, and Internet of Things (IoT) networks. However, due to their decentralized nature and lack of centralized authority, these networks are open to different types of security threats, among which the Sybil attack is one of the most challenging and disruptive.

The Sybil attack, introduced by Douceur [1] in peer-to-peer distributed systems, represents a malicious node that creates multiple false identities to gain unwarranted control of the network. By masquerading as multiple legitimate nodes, the attacker can manipulate communications, disrupt routing, and compromise data integrity. Consequently, it is essential to detect and mitigate Sybil attacks in MANETs in order to maintain the security and reliability of the network.

Several solutions have been proposed for the detection of Sybil attacks, each with its advantages and disadvantages. Unfortunately, most of these solutions are not suitable for MANETs. Cryptographic methods are generally resource-intensive, while position-based techniques require specialized hardware. At the same time, solutions based on signal strength or those based on monitoring nodes introduce constraints that are unsuitable for decentralized and mobile ad hoc networks. This highlights the critical need for the design of detection methods better matched to the MANET setting.

The main contributions of this research are:

- Mobility-Correlation-based Detection: Sybil identities detection through mobility-pattern correlation derived from neighbor-list dynamics instead of relying on physical-layer metrics such as RSSI or location information.
- Fully Distributed Architecture: Unlike previous mobility-context or watchdog-based approaches, this method performs a distributed detection without synchronized observers or dedicated monitoring nodes, making it ideal for MANETs.
- Dual Threshold Mechanism: To optimize detection accuracy, we have introduced a dual threshold mechanism in order to balance the precision and the recall, which ensures maximal accuracy with minimal false alerts.
- Secure Lightweight Dissemination: To ensure secure sharing of detection results between nodes against attackers' false alerts, we used the QUARK lightweight hash function, which introduces negligible communication overhead.
- Whitewashing Mitigation: New nodes in the neighbor list are undergoing a probation period (limited routing privileges, rate limits, no vote in consensus) to mitigate whitewashing attempts.

All these features make the method highly efficient in terms of resource consumption and well suited to mobile infrastructure-less MANETs.

The importance of this work is justified by the increasing use of mobile ad hoc communication in various fields such as military operations and disaster relief operations, where

Manuscript received December 10, 2025; revised May 6, 2026. Date of publication August 17, 2026. Date of current version August 17, 2026. The associate editor prof. Toni Perković has been coordinating the review of this manuscript and approved it for publication.

S. Selmane is with the Research Unit LaMOS, Faculty of Exact Sciences, Department of Computer Science, University of Bejaia, Algeria (e-mail: samir.selmane@univ-bejaia.dz).

A. Baadache is with the Research Unit LaMOS, Department of Computer Science, University of Algiers, Algeria (e-mail: a.baadache@univ-alger.dz).

F. Semchedine is with the Department of Computer Science, University of Setif, Algeria (e-mail: fouzi.semcheddine@univ-setif.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0259

network infrastructure is unavailable or expensive and node mobility is high. This requires the design of a lightweight and scalable Sybil attack detection method, and the proposed method offers a practical and effective solution.

The rest of this article is structured as follows: Section II presents the state of the art of work related to the detection of Sybil attacks. Section III introduces the assumptions of the proposal, the attack model, and the notation used. Section IV then details the proposed method, followed by the simulation results and comparison in Section V. Finally, Section VI concludes the article.

II. Related Works

Sybil attacks pose a severe threat to the integrity and reliability of MANETs. Several solutions have been proposed for the detection of this attack using various techniques. This section categorizes and reviews the proposed approaches to detect Sybil attacks in MANETs.

There have been some works that investigate the exploitation of Received Signal Strength (RSS) inconsistencies to detect Sybil nodes. Abbas et al. [2] differentiate legitimate newcomers from Sybil identities based on their RSS variations at network entry, utilizing a threshold based on speed and RSS fluctuations. Yao et al. [3] propose "Voiceprint", a radio propagation model-free, distributed approach that analyzes RSSI time series for correlated signal patterns and employs the Bernaola Galván Segmentation Algorithm (BGSA) to identify abrupt changes in RSSI time series, which helps detect illegitimate nodes that manipulate their transmission power. Paul and Sinha [4] employ Bezier curves to analyze Received Signal Power (RSP) patterns, and use the tangent properties of these curves to distinguish between legitimate and Sybil nodes. Vadhana et al. [5] propose authentication for anonymous location-based routing (AALBR), which classifies nodes into safe or caution zones based on RSS and arrival angles. Group signature schemes are used for authentication, and Ant colony optimization-based route discovery is employed to prevent misrouting attacks. Chen et al. [6] leverage the RSS and K-means clustering to detect Sybil attackers, consider power control manipulation while integrating real-time indoor positioning for attacker localization. Mulla and Sambare [7] propose a method based on analyzing entry and exit behavior through RSS variations, utilizing a threshold-based detection mechanism and historical RSS data tracking. Abbas et al. [8] propose a one-time localization scheme, minimize overhead by localizing new nodes while entering the network, store verified identities in a Registered Identity List (RIL) and use multi-neighbor validation to avoid collusion. Guven and Taysi [9] created a dataset for the detection of Sybil nodes in vehicular ad hoc networks (VANETs) based on the RSSI. This is done by developing a large-scale simulation environment by using a log-normal shadowing model to simulate signal attenuation and fluctuations. In another work, Almesaeed [10] proposed a technique for Real-Time Sybil Classification and Detection (RTSCD). This method integrates 3D channel characterization with the RSSI, and uses Dynamic Time Warping to distinguish legitimate vehicles from Sybil ones.

Location-Based Verification utilizes location information and traffic patterns in order to detect any anomalies created by Sybil nodes. Ayaida et al. [11] leverage traffic flow theory and collaborative verification to discover Sybil nodes by comparing actual vehicle speeds against expected speeds. Zhang et al. [12] analyze basic security message (BSM) packets to identify unique sources by comparing calculated and claimed transmission distances, detecting Sybil attacks based on mismatches. Tangpong et al. [13] present a collaborative-based distributed approach that uses hop-by-hop packet signatures and periodic observation exchanges for the reconstruction of node paths and detecting Sybil attackers with similar movement patterns. Bouassida et al. [14] propose a localization verification scheme, comparing predicted and actual RSS values, and utilizing "distinguishability degree" evaluation to detect Sybil attack. Rajadurai and Gandhi [15] propose Fuzzy based collaborative verification system (FCVS). This approach utilizes stable nodes as monitoring nodes for checking location claim and employs Fuzzy logic for categorizing nodes according to the difference in distance, angle, and RSSI. Piro et al. [16] propose a passive approach that monitors network traffic to identify potential Sybil attackers by analyzing the co-occurrence patterns of transmitting node identities. A recent study by Khatri et al. [17] introduced a blockchain-based proof-of-location mechanism. This mechanism exploits the hardware constraints of vehicles' electronic control units (ECUs) by sending computational challenges that a single attacker cannot solve, and stores validated locations on the blockchain to ensure data integrity.

Trust and reputation-based systems are deployed to create trust relations among nodes and detect malicious activities. Fallah and Mouzarani [18] model MANET interactions as a multi-stage game with a reputation system that imposes costs for maintaining multiple identities, discouraging Sybil attacks. Ubarhande et al. [19] enhance the IDS-AODV protocol using timestamps to validate observer node reports, enabling the identification and quarantine of black hole and gray hole attackers. Bharti et al. [20] propose a Two-Tier Multi-Trust-Based Algorithm to Countermeasure the Sybil Attacks, a two-tier trust system, with a proficient cluster head selection algorithm and a hybrid routing approach to mitigate Sybil attacks and optimize network performance. Abbas et al. [21] introduce a mechanism that forces new nodes to contribute to the network before accessing services, thereby discouraging Sybil attacks. Bochem et al. [22] propose "Unchained", a blockchain-based identity system that requires a cryptocurrency deposit to create verifiable identity proofs, discouraging Sybil identities. Another work by Hassan et al. [23] proposed LETM-IoT, a centralized trust mechanism for IoT networks. This method compares the source, parent, and sequence identifiers with a valid network topology database to detect any manipulations and blacklists Sybil nodes. Stepien et al. [24] propose the B&STL&T and B&SEBP&F algorithms that use trust level evaluation, timestamp verification, and trajectory-based footprint tracking to detect bogus and Sybil attacks in VANETs.

Bio-inspired algorithms take inspiration from the natural and biological world to design new mechanisms for detecting

Sybil attacks. Memarmoshrefi et al. [25] introduce a detection mechanism based on social insects that analyzes the consistency of reported identities within exchanged certificate chains. Mahajan et al. [26] make the Dynamic Source Routing (DSR) protocol more resistant against Sybil attacks by integrating the Bacterial Foraging Optimization (BFO) algorithm to optimize routing paths. Rao et al. [27] propose an adaptive network-based fuzzy inference system model, enhanced with a Bloom Filter, to predict Sybil attacks. Additionally, Iwendi et al. [28] propose a new approach named SMTS (Spider-Monkey Time Synchronization) for large-scale VANETs to detect Sybil attacks. This technique aims to enhance packet delivery time synchronization while minimizing energy consumption. Recent research [29] has proposed a framework that uses Glow Worm Swarm Optimization (GSO) for clustering stabilization and Elephant Herding Optimization (EHO) enhanced with Chaotic Maps to optimize a CNN model used to detect Sybil Attacks.

Neighboring Node Information-based methods investigate neighboring node information to identify Sybil identities. Jamshidi et al. [30] use Watchdog Nodes (WNs) to monitor network traffic and node mobility, attribute bitwise labels based on movement patterns to identify Sybil nodes. This approach identifies Sybil nodes by grouping identities with the same bit-labels, assuming Sybil identities from a single physical device will move in the same way. Almas-Shehni and Faez [31] propose a lightweight algorithm leveraging node mobility context. This method also uses WNs to track node co-occurrence patterns in a matrix (M_occur) over multiple iterations. A Sybil node is discovered if there are a certain number of IDs that frequently occur jointly in the neighborhood of WN. Ssu et al. [32] propose a solution that identifies Sybil nodes based on the consistent presence of ”critical members” in their common neighbor sets. Their idea exploits that all forged Sybil nodes from the same physical device will share the same set of neighbors. The approach does not require any form of authentication, location information or dedicated hardware. Grover et al. [33] analyze the intersection of periodically exchanged neighbor lists in VANETs to detect nodes appearing together for significant durations as potential Sybil nodes. Their distributed approach assumes that physically distinct legitimate vehicles cannot have identical neighbor sets for a long period, unlike fake identities from a single attacker. Sharma and Singh [34] present a clustering-based approach that examines neighbor lists exchanged through Hello packets to discover fake identities that are moving together. Their ”Detector” node compares neighbor lists across multiple time instances, performing set intersections to find groups of identities that consistently share common neighbors. Kalaiselvi and Sundaram [35] proposed a detection strategy based on the AdaBagging algorithm. This method extracts characteristics such as the degree of the node, the average degree of the neighbor, and the grouping coefficients to classify the nodes. Subsequently, they introduced a ”Unified Framework” [36] that integrates AdaBagging with the Regressive Arboretum Ensemble (ERA). AdaBagging gives an initial classification and ERA calculates risk scores based on mobility and frequency of communication to optimize the initial classification.

Despite the extensive research on Sybil attack detection, current solutions are confronted with major limitations in terms of scalability, overhead, and adaptability to highly mobile MANET environments. RSS and geolocation-based solutions, while effective in static or infrastructure networks, suffer from signal fluctuations and reliance on specific hardware. However, they also require trusted entities, which can pose security risks and make them unsuitable for decentralized mobile networks. Trust- and reputation-based systems, on the other hand, generate significant computational and communication overhead. Similarly, blockchain and identity management methods pose constraints of permanent connectivity and availability of resources that are not feasible in the context of MANETs.

Although the proposed neighbor-based Sybil detection techniques are lightweight, they suffer from several factors limiting their feasibility in dynamic MANETs. Approaches such as those in [30], [31], and [34] rely on centralized monitoring nodes or require clustered topologies, which present a single point of failure or increase the communication and coordination burden between nodes in the network. Other works, such as those of [30], [32], and [33], rely on the nodes' honesty cooperation in sharing neighboring sets or similar movement patterns, making them highly susceptible to subtle Sybil manipulations and signal fluctuations in mobile, less dense networks. Computational overhead can also be significant, in some methods like [34] having $O(n^3)$ complexity. On the other hand, most methods do not address the problem of whitewashing attempts and do not include techniques for secure dissemination of detection results, which poses the risk of false alerts. All these problems show the need for distributed, lightweight detection and mitigation methods adapted to the context of MANETs.

To address these limitations, we have proposed a lightweight, fully distributed, neighbor-based Sybil attack detection and mitigation method. This method is not only effective but also highly scalable. The method exploits mobility correlation between nodes to identify groups of identities that move together, an inherent characteristic in the behavior of Sybil nodes. It integrates a dual-threshold decision mechanism to balance precision and recall. To securely disseminate the detection results without exposing node identities, and minimize the risk of false alerts, we used the QUARK lightweight hash function. In order to reduce whitewashing attempts, new nodes arriving in the neighbor list are subject to a probation period. Unlike prior works, the proposed approach requires no additional infrastructure, cryptographic key management, or monitoring coordination, making it both scalable and energy-efficient for real MANET deployments.

III. System Assumptions, Attack Model and Symbols

Our scenario involves a fully distributed, ad hoc network of mobile nodes communicating wirelessly. There is no pre-existing infrastructure or centralized authority. Sybil attack detection is fully distributed, where each node is responsible for detecting malicious behavior. Newsome et al. [37] established a Sybil attack taxonomy based on three orthogonal

dimensions: direct vs. indirect communication, fabricated vs. stolen identities, and simultaneity. Our detection method focuses on the direct, fabricated, and simultaneous attack variant, as defined within this framework.

- sequence: represents the sequence of appearance and disappearance timestamps of a node in the neighbor list of another node.
- sequences_list: a list containing all the sequences of nodes in the neighbor list.
- LCS: represents the length of the Longest Common Substring between two sequences. Given are two sequences S_1 and S_2. The task is to find the length of the longest common substring between S_1 and S_2. For example: S_1 = "In1Out2In4Out6In7Out8", S_2 = "In1Out3In4Out6In7Out9". The longest common substring of these two token sequences is "In4Out6In7", so the LCS is 3 (three matching tokens in a row).
- $LCS_{baseline}$: the maximum LCS value observed between any pair of legitimate nodes during a no-attack calibration phase. This is the empirical baseline used to set the two thresholds defined below.
- threshold_min: the LCS value at or above which two identities are considered potential Sybil identities. We set $threshold\_min = LCS_{baseline}$.
- threshold_max: the LCS value at or above which two identities are considered confirmed Sybil identities. We set $threshold\_max = LCS_{baseline} + \delta$, where δ is an offset chosen empirically to eliminate false positives.
- hash_value: the hash value of the identity set of a confirmed Sybil node.
- hash_list: the list of hash values received from other nodes.
- suspect_list: the list of suspected Sybil nodes together with the hash value of their identities.
- blacklist: the list of confirmed Sybil nodes.

IV. Detection Method

The lack of infrastructure and centralized authority in MANETs requires autonomous, distributed threat detection. Our proposed method is fully decentralized where each node independently identifies Sybil attacks by leveraging the synchronized mobility characteristics of Sybil identities. Figure 1 illustrates the steps to detect and mitigate the Sybil attack. These steps are performed by each node in the network.

A. Core Principle: Mobility Correlation

Each node continuously monitors its immediate neighborhood list in order to construct a chronological sequence of appearance and disappearance timestamps for each neighbor. The main idea is that Sybil identities belonging to the same physical device will have strongly correlated, often identical appearance/disappearance sequences. So, the detection method will identify similar sequences, indicating potential Sybil identities.

B. Threshold Determination for Robust Detection

The proposed dual-threshold mechanism uses two thresholds, $threshold\_min$ and $threshold\_max$, which have different but complementary roles. When the LCS value of a pair of identities reaches $threshold\_min$, the pair is considered suspicious and its communication is restricted, but it is not yet confirmed as a Sybil pair. When the LCS value reaches $threshold\_max$, the pair is immediately blacklisted as a confirmed Sybil pair.

The two thresholds are designed to optimize different objectives. $threshold\_min$ is selected to ensure that all real Sybil pairs are detected as suspicious, which maximizes recall. In contrast, $threshold\_max$ is selected to ensure that legitimate pairs are never blacklisted, which maximizes precision. The interval between the two thresholds forms a soft-decision zone where the dissemination mechanism described in Section IV-E can provide additional evidence.

Both thresholds are obtained during a calibration phase performed without attackers. The calibration procedure is described below:

1) Run K simulations in normal conditions without attacks while varying the deployment parameters shown in Table III, including the number of nodes, network area, communication range, and node speed.
2) For each simulation and each observer node, compute the LCS of the appearance and disappearance sequences for every pair of nodes in the neighbor list. The LCS value is limited to twice $threshold\_max$, as explained in Section IV-C.
3) Collect all LCS values from all observers and simulations. Define $LCS_{baseline}$ as the maximum value in this distribution. $LCS_{baseline}$ represents the longest synchronized neighbor-list sequence that can occur naturally between legitimate nodes under the tested conditions.
4) Set $threshold\_min = LCS_{baseline}$. Any pair whose LCS exceeds this value cannot be explained by normal independent mobility and is therefore considered suspicious.
5) Set $threshold\_max = LCS_{baseline} + \delta$. The offset δ is used to absorb remaining variability that may not be captured by the maximum baseline value. This ensures that legitimate pairs are not incorrectly blacklisted even in unusual mobility situations.

The LCS computation uses the Suffix Tree/Suffix Array approach because it provides optimal linear complexity in time and space, $O(n + m)$ [38]. This makes the method efficient for sequence comparison.

The value of δ is determined experimentally rather than analytically. Section V-B (Figure 4) shows the precision of the dual-threshold mechanism for different values of δ. The results show that $\delta = 3$ is the smallest value that achieves stable 100% precision across all tested iteration counts. Therefore, $\delta = 3$ is used throughout this paper.

Smaller values of δ maintain high precision in most cases but may produce occasional false positives when the number of iterations is low. Larger values also achieve high precision, but they increase the delay before a Sybil pair is fully con-

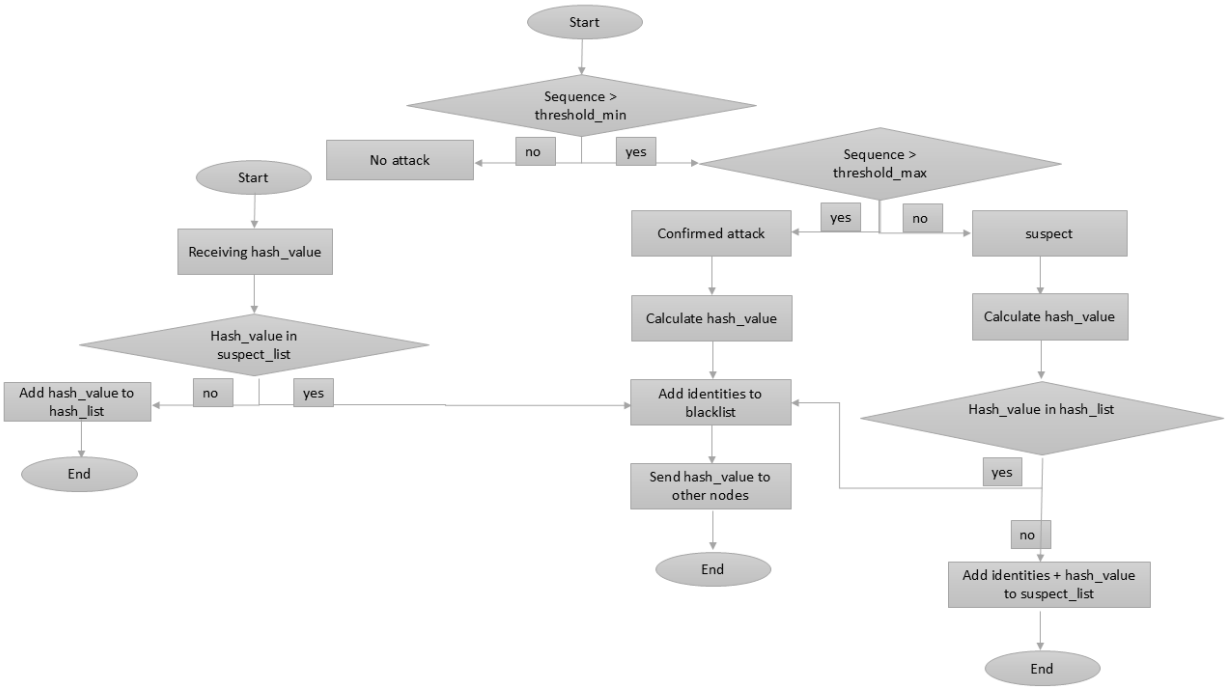


Fig. 1. Proposed Method Flow Chart

firmed. Thus, δ acts as a configurable parameter that balances detection speed and tolerance to false alarms.

Although the calibration phase described above is performed offline, the same procedure can also be applied online. In this case, $LCS_{baseline}$ can be periodically recalculated using a sliding window of recent neighbor-list traces collected from the deployed network. This allows the system to adapt to gradual changes in node mobility and network density. In stable deployments where the operating conditions are already known, a single offline calibration is sufficient.

C. Computational Efficiency

To minimize the computational overhead, and since the complexity of the LCS computation based on the Suffix Tree/Suffix Array method is O(n+m), we have limited the length of the sequences used for the comparison to twice the value of the threshold_max. This limit will ensure a balance between detection accuracy and computational efficiency, which ensures that the complexity remains linear relative to the size of the neighbor list. This linear complexity makes the method suitable for resource-constrained mobile devices.

D. Dual-Threshold Decision

Our detection algorithm (Algorithm 1) is based on the following decisions:

- Less than threshold_min: If the length of the common sequence is less than threshold_min, the node takes no action because the correlation is not yet significant and is probably legitimate.
- Between threshold_min and threshold_max: If the length of the common sequence is between the two thresholds, the correlation is considered suspicious. The node will perform an internal check: it will look for the hash of these identities in its hash_list (received from other nodes). If the hash is found, it confirms a Sybil attack. Otherwise, the node will add the suspicious identities and their hash values to a suspect_list. It will impose restricted communication on these identities (e.g., limited routing privileges, frequency limits, lack of consensus voting) to limit immediate risks.
- At or above threshold_max: If the length of the common

Algorithm 1 Detection algorithm

$blacklist[]$ ▷ contain confirmed Sybil identities
$suspect\_list[]$ ▷ contain non-confirmed Sybil identities+their hash_values
$hash\_list[]$ ▷ contain hash_values received from other nodes
if $(sequence \geq threshold\_max)$ **then**
  - calculate hash_value of identities
  - add identities to blacklist
  - send hash_value to other nodes
else
  if $(sequence\ is\ between\ threshold\_min\ and\ threshold\_max)$ **then**
    calculate hash_value of identities
    if $(hash\_value\ is\ in\ hash\_list)$ **then**
      - add identities to blacklist
      - send hash_value to other nodes
    else
      - add identities and hash_value to suspect_list
    end if
  end if
end if

sequence is at or above threshold_max, a confirmed Sybil attack is detected. The node will blacklist the detected Sybil identities and completely block all communication with them.

E. Secure Dissemination and Whitewashing Mitigation

If a Sybil attack is confirmed (either by exceeding the threshold_max or by exceeding the threshold_min and receiving the hash value), the detector node will share the information with other nodes. To prevent the spread of false alerts (attacker disseminates legitimate identities as Sybil identities), it will transmit the hash of the attackers' identities instead of the real identities. This hash is computed using the QUARK lightweight hashing function; a lightweight hashing function designed and optimized for resource-constrained devices [39]. We chose QUARK for its minimal computational load and its ultra-low power consumption, which is ideal for MANETs.

When a node receives a hash value (Algorithm 2), it will check its suspect_list, which contains the suspicious hash values. If the received hash is in this list, it will move those identities from suspect_list to the blacklist. Otherwise, the hash is added to the node's hash_list for future verification. This hash-based technique prevents attackers from disrupting the detection process by disseminating false alerts.

Algorithm 2 Receiving hash_value

$blacklist[]$ ▷ contain confirmed Sybil identities
$suspect\_list[]$ ▷ contain non-confirmed Sybil identities+their hash_values
$hash\_list[]$ ▷ contain hash_values received from other nodes
if ($hash\_value\ is\ in\ suspect\_list$) **then**
- add identities to blacklist
- send hash_value to other nodes
else
- add hash_value to hash_list
end if

Moreover, in order to decrease the effectiveness of whitewashing attempts, any new identity in the node's neighbor list is initially placed in a probation period. During this period, this node will have limited routing privileges, rate limits, and no consensus voting rights until it demonstrates a non-suspicious appearance/disappearance sequence, which proves its legitimacy.

F. Illustrative Example

To illustrate how our Sybil attack detection method works in practice, we considered a network of seven nodes, numbered 1 to 7. In our scenario, node 4 is a Sybil attacker, with two identities: 4 and 8. Figure 2 chronologically illustrates the dynamic changes in node positions from time T_1 to T_8.

To show the detection process, we will examine the lists of neighbors of Node 3 in all the temporal instances (T_1 to T_8). Table I shows Node 3's neighbor lists from T_1 to T_8.

TABLE I
NODE3'S NEIGHBOR LISTS FROM T_1 TO T_8

Time Ti	Neighbors' list
T_1	{1,4,8}
T_2	{4,6,8}
T_3	{5,6}
T_4	{1,6}
T_5	{2,7}
T_6	{2,4,8}
T_7	{5,6,7}
T_8	{4,5,8}

To detect Sybil attacks, Node 3 will construct a chronological sequence for each node in its neighboring list. This sequence will contain the timestamps of appearance ("In") and disappearance ("Out") of Node 3's neighborhood. For example, Node 5 has the sequence "In3Out7", which indicates that it appeared in the Node 3 neighbor list at timestamp T_3 and disappeared at timestamp T_7.

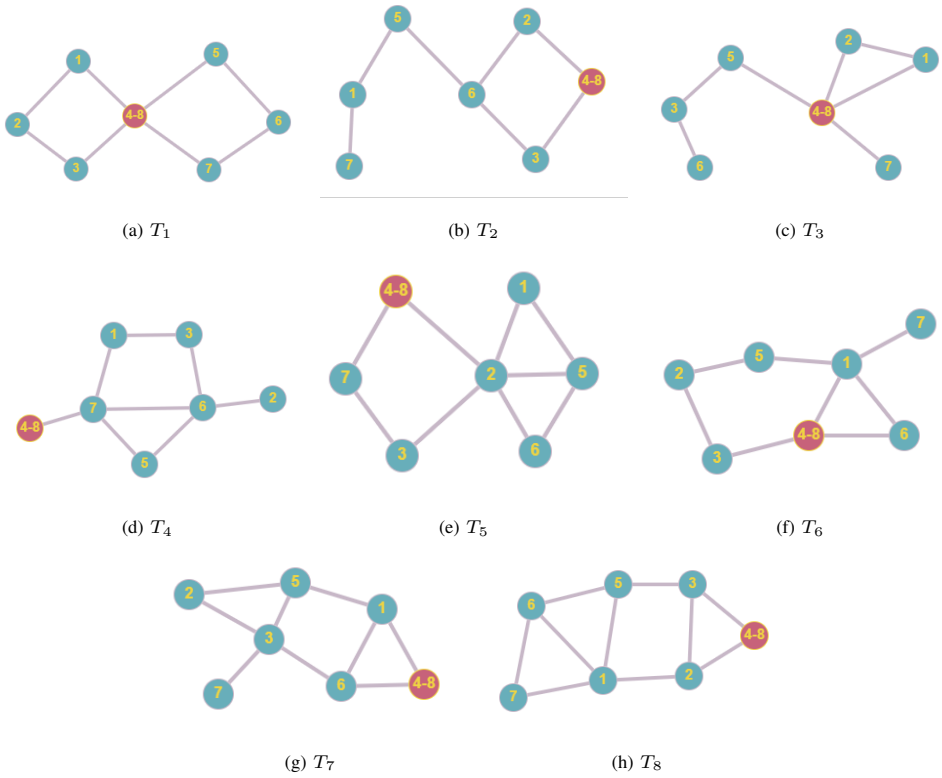
Table II shows the sequences computed by node 3 for each of the nodes in its neighborhood list. From Table II, it is obvious that node 4 and node 8 have the same appearance/disappearance sequence: "In1Out3In6Out7In8". This strong correlation indicates a potential Sybil attack, as physically distinct legitimate nodes cannot have such identical long sequences.

TABLE II
NODE3'S SEQUENCES FOR EACH NODE

Time Ti	Sequence
1	In1Out2In4Out5
2	In5Out7
4	In1Out3In6Out7In8
5	In3Out7
6	In2Out5In7Out8
7	In5Out6In7Out8
8	In1Out3In6Out7In8

To make a definitive decision, node 3 will compare the length of the common sequence with the threshold_min and threshold_max values.

- If the length of the common sequence is less than threshold_min, Node 3 does nothing because there is insufficient correlation to confirm an attack.
- If the length of the common sequence is at or above threshold_max, this confirms a Sybil attack. Node 3 will blacklist identities 4 and 8, calculate their hash value using the QUARK function, and disseminates that hash to the other nodes in the network.
- If the length of the common sequence is between threshold_min and threshold_max: this indicates a suspicious correlation. Node 3 will compute the hash value of identities 4 and 8. It will then check if this hash already exists in its hash_list (this means that another node has already confirmed this attack and shared the hash).
  - If the hash exists in the hash_list: This will validate the suspicion and confirm the Sybil attack. Node 3 will add identities 4 and 8 to the blacklist and share their hash_value with the other nodes.
  - If the hash does not exist in the hash_list: Node 3 will append the identities 4 and 8 and their hash value, to



(a) T_1 (b) T_2 (c) T_3

(d) T_4 (e) T_5 (f) T_6

(g) T_7 (h) T_8

Fig. 2. Network topologies

the suspect_list. It will then apply communication restrictions (e.g., limited routing privileges, frequency limits, lack of consensus voting) to these suspicious identities, and will wait for additional validation by other nodes or by exceeding the threshold_max to confirm the attack.

In either case (whether an attack is confirmed or identities are merely deemed suspicious) these actions result in the isolation of nodes 4 and 8 from the network, thus limiting their impact.

V. Simulation Results and Discussion

In this section we will evaluate the efficiency, effectiveness, and robustness of our proposed Sybil attack detection method using the INET framework under the OMNeT++ simulator.

A. Threshold Determination and Characterization

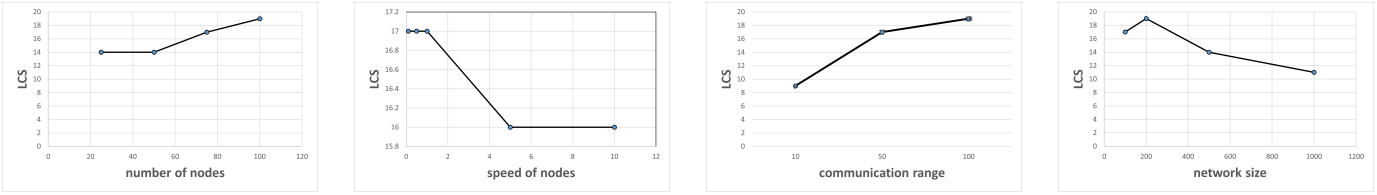
The first step is to determine the optimal values for our dual-threshold mechanism: threshold_min and threshold_max. In order to balance detection sensitivity and false positive rates, these thresholds will define the minimum common sequence length required for two nodes to be considered potential or confirmed Sybil identities.

To determine these values, we simulated the network several times under different parameters in normal conditions (no-attack). During these simulations, each node will compute the LCS of the sequences of appearance/disappearance between each peer of its neighbors. Variations of these LCS values with different network conditions help us define the maximum sequence lengths for legitimate colocation. Table III shows the network parameters used in the threshold determination process.

TABLE III
Simulation Parameters

Parameter	Values
number of nodes	25, 50, 75, 100
network size	100*100m, 200*200m, 500*500m, 1000*1000m
node's speed	0.1mps, 0.5mps, 1mps, 5mps, 10mps
node's communication range	10m, 50m, 100m
node's mobility	Random Way Point

Figure 3 illustrates how the average maximum length of LCS observed between legitimate nodes (which determines our thresholds) correlates with varying network parameters. We observed that this length is directly proportional to the number of nodes and the communication range, because a higher density and a wider range increase the probability of longer common sequences. Conversely, it is inversely proportional to the size of the network and the speed of the nodes, because larger areas and faster mobility break up the long commonalities between the neighborhoods. This analysis characterizes how $LCS_{baseline}$ depends on the operating conditions of the MANET. The calibration procedure of Section IV-B can therefore be applied online (re-derived from a sliding window of observed neighbor-list traces) whenever the network's mobility or density characteristics drift, ensuring that the thresholds remain accurate without manual retuning.



(a) number of nodes (b) speed of nodes (c) communication range (d) network size

Fig. 3. maximum sequence

B. Performance Evaluation

After the threshold determination, we evaluated the performance of the proposed method in various attack scenarios. In these simulations we varied the number of Sybil attackers and the number of identities per attacker, as well as the number of iterations. Table IV presents the parameters used for the performance evaluation simulations.

TABLE IV
DETECTION SIMULATION PARAMETERS

Parameter	Values
number of attackers	2, 3
number of identities per node	2, 3
number of iterations	50 to 500
threshold value	LCS to LCS+5

We evaluated the performance of our method using precision, recall, and accuracy metrics:

Precision: It represents the ratio of correctly detected Sybil nodes to all detected ones. Figure 4 shows the proportional relationship between the precision, the number of iterations, and the threshold value. We noted that as long as the threshold is LCS+3 or higher, the precision is 100%, regardless of the number of iterations. This indicates that a threshold value of LCS+3 classifies a node as Sybil with no false positives.

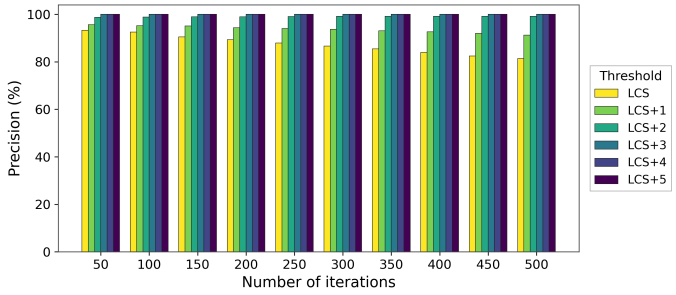


Fig. 4. Precision

Recall: The recall measures the proportion of actual Sybil nodes that have been correctly identified. Figure 5 shows that the recall increases proportionally with the number of iterations but decreases as the threshold increases. We noticed that 100% recall is achieved with 350 iterations or more even with lower thresholds, which suggests that a sufficient number of iterations is needed to detect all malicious nodes.

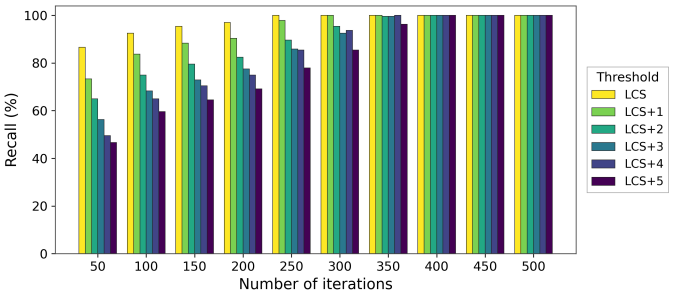


Fig. 5. Recall

False positive rate: As observed in the precision, Figure 6 shows a high decrease in false positives with respect to the number of iterations and the threshold. It is clear that a threshold of LCS+3 or higher eliminates false positives, which makes the method more effective in preventing misclassification of legitimate nodes.

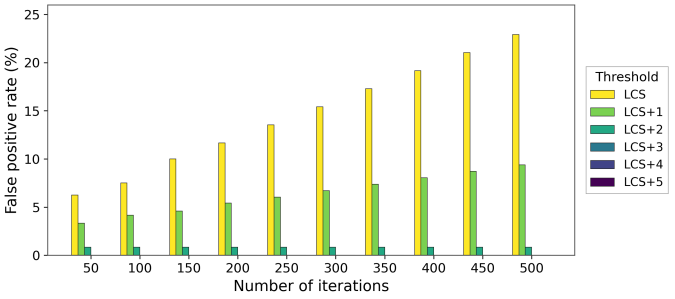


Fig. 6. false positive

Accuracy: Figure 7 summarizes these results, showing that the overall accuracy reaches up to 100% when the number of iterations is 350 and the threshold is between LCS+2 to LCS+4.

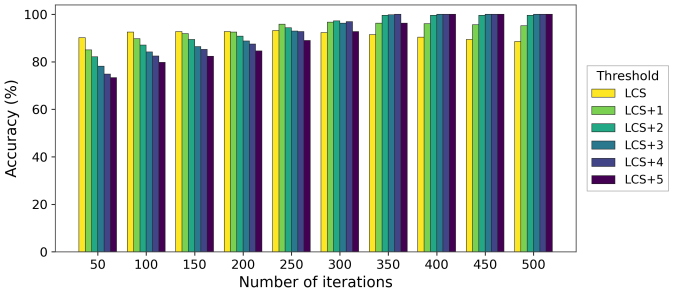


Fig. 7. Accuracy

TABLE V
RESULTS OF OUR SYBIL ATTACK DETECTION METHOD

Threshold	Precision	Recall	Accuracy	False positive	Detection rate	
					Before Communication	After Communication
threshold_min	85,41%	100%	91,46%	17,29%	83,88%	100%
threshold_max	100%	99,58%	99,79%	0%	66,29%	100%

C. Optimal Threshold Settings

Based on these overall findings, we would suggest threshold_min = LCS (the LCS value from non-attack patterns) and threshold_max = LCS + 3, with a minimum number of 350 iterations. This setup achieves the maximum overall accuracy: 100% perfect recall (as threshold_min will, in practice, identify all potential Sybil nodes) and 100% perfect precision (since threshold_max eliminates false positives). Table V summarizes these optimal performance metrics.

D. Impact of Shared Awareness

Figure 8 shows the advantage of our mechanism of securely disseminating detection results. It shows a large increase in the average number of nodes detecting the attack after reporting the hash values of confirmed Sybil identities. This underscores the important role of shared awareness in quickly gaining wide network consensus against attackers, thereby isolating Sybil attackers and improving the overall effectiveness of the detection method.

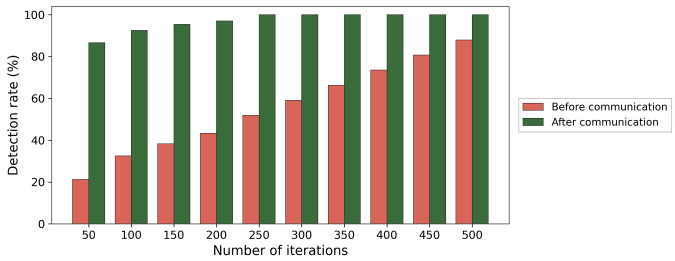


Fig. 8. average detection rate

The results in Table VI prove the superior performance of our method. It achieves 99.8% accuracy and 0% false positives with minimal communication overhead, which distinguishes it from traditional methods that either rely on RSSI measurements, require directional antennas, require specialized infrastructure, or have cryptographic overhead. It should be noted that our method is infrastructure-independent and excels in highly dynamic environments by taking advantage of mobility.

E. Whitewashing Mitigation Effectiveness

To mitigate whitewashing attempts, in which malicious devices discard a tagged identity and reappear under a new one, we implemented a lightweight probation and linkage policy. Each newly observed identity is placed in a probationary state for a short observation window (corresponding to threshold_min iterations), during which its participation privileges are limited. At the same time, nodes compute behavioral similarity between the new identity and any recently broadcast hashed suspect identities using their existing neighbor list. If similarity is detected, the new identity is flagged as suspected and isolated. This approach adds negligible computational overhead and effectively prevents rapid identity whitewashing without the need for physical-level fingerprinting or centralized infrastructure. This low overhead means that the system remains resource-efficient.

F. Cross-Simulator Validation

To confirm that the results are not specific to one simulator and to evaluate the generality of the proposed method, we reimplemented the detection approach in NS-3 version 3.44. The detection algorithm itself is independent of the simulator because it only uses neighbor-list dynamics. Therefore, the detection logic remained unchanged, while only the simulation environment was modified. NS-3 generated neighbor-list snapshots that were processed using the same detection code.

The evaluation was performed in a 500×500 m area with a 100 m communication range using a range-limited propagation model. The number of nodes was varied over $\{25, 50, 75, 100\}$. Unlike the constant-speed model used in Section V-A, each node selected its speed independently at every waypoint from Uniform$[1, 10]$ m/s. This created a heterogeneous mobility scenario closer to realistic network conditions. For each configuration, a 1000 s attack-free simulation was first used to calculate $LCS_{baseline}$. Then, a 350 s simulation with 2 attackers (2 identities per attacker) was evaluated using $threshold\_min = LCS_{baseline}$ and $threshold\_max = LCS_{baseline} + 3$, following the method described in Section IV-B.

Table VII presents the comparison between the two simulators using the same format as Table V. At $threshold\_max$, the results from OMNeT++/INET and NS-3 are very similar. Both simulators achieved 100% precision and recall, accuracy between 99.79% and 100%, a 0% false positive rate, and full network-wide detection after hash dissemination.

At $threshold\_min$, both simulators still achieved perfect recall and reached 100% detection after dissemination. However, NS-3 produced a much lower false positive rate (1.56% compared to 17.29%), which resulted in higher accuracy (98.44% compared to 91.46%). The lower precision observed in NS-3 (47.47% compared to 85.41%) does not contradict these results. Precision depends on the value of $LCS_{baseline}$, which is used to determine threshold_min. The value of $LCS_{baseline}$ is influenced by the mobility, propagation, and connectivity models used by the simulator. In the NS-3 experiments, the calibration phase produced relatively small $LCS_{baseline}$ values. As a result, threshold_min is set close to the noise level, which causes the method to classify even short co-encounters between node pairs as suspicious. In contrast, Figure 3 shows that the OMNeT++/INET evaluation produced larger $LCS_{baseline}$ values. Therefore, threshold_min is placed

TABLE VI
COMPARISON OF OUR SYBIL ATTACK DETECTION METHOD WITH RELATED WORKS' METHODS.

Ref	Detection Technique	Accuracy (%)	False Positive Rate (%)	Overhead/Requirement	Simulator
Proposed Method	Neighbor-based	>99% (up to 100%)	<0.5% (down to 0%)	Minimal communication overhead + No crypto infrastructure/specialized hardware.	OMNeT++ (INET framework) + NS-3 (cross-validation)
[2] Abbas et al.	RSS-based	∼90% TP	∼10% FP	RSS fluctuation	Java Sun SPOT sensors (testbed)
[3] Yao et al.	RSS-based	∼100%	∼0%	VANETs + RSS fluctuation	Extensive simulations & real-world testbed
[4] Paul & Sinha	RSS-based	∼90%	∼10%	RSS fluctuation	NS2
[6] Chen et al.	RSS-based	>95%	<5%	WSN + No mobility + Localization system	WiFi and ZigBee networks
[10] Almesaeed.	RSS-based	>90%	N/A	VANETs + RSS fluctuation	MATLAB
[11] Ayaida et al.	Location-based	>90%	N/A	VANETs + Vehicle speed calculation + V2V communications	NS3 and SUMO simulators
[12] Zhang et al.	Location-based	>98%	N/A	VANETs + BSM manipulation	F2MD framework
[13] Tangpong et al.	Location-based	>80%	∼10%	Requires directional antennas + uses signatures for traffic observation	NS2
[16] Piro et al.	Location-based	∼100%	∼0%	Passive approach + single/multi-observers	NS2
[23] Hassan et al.	Trust-based	100% TP	N/A	IoT + No mobility + Centralized	Cooja (Contiki OS)
[29] Suganyadevi et al.	Bio-inspired	98.9%	2%-8% FN	VANETs + Cryptography	NS-2.35
[30] Jamshidi et al.	Neighbor-based	>94%	0%	Relies on Watchdog Nodes	J-SIM
[31] Almas-Shehni & Faez	Neighbor-based	∼99%	<2%	Relies on Watchdog Nodes.	J-SIM
[32] Ssu et al.	Neighbor-based	>99%	<4%	WSN + No mobility + Node detector specific requirements	J-SIM

TABLE VII
CROSS-SIMULATOR COMPARISON.

Simulator	Threshold	Precision	Recall	Accuracy	False positive	Detection rate	
						Before Comm.	After Comm.
OMNeT++/INET	threshold_min	85.41%	100%	91.46%	17.29%	83.88%	100%
	threshold_max	100%	99.58%	99.79%	0%	66.29%	100%
NS-3	threshold_min	47.47%	100%	98.44%	1.56%	65.21%	100%
	threshold_max	100%	100%	100%	0%	19.53%	100%

much higher than the noise level, which naturally filters out most false matches.

The difference in the "Detection rate before communication" column is also explained by the same network configuration. In the larger 500×500 m NS-3 environment, each attacker is directly observed by a smaller percentage of nodes. However, the hash dissemination mechanism described in Section IV-E allows both simulators to achieve 100% network-wide detection after communication.

The strong agreement at $threshold\_max$ across two independent simulators shows that the proposed method is not dependent on the OMNeT++/INET implementation. In addition, the results at $threshold\_min$ confirm that the dissemination mechanism of Section IV-E provides the same balancing effect in both simulation environments.

Finally, the evaluation presented here is limited to the operating conditions for which the method was designed: a network area that is neither too small nor too large, a communication range comparable to the average inter-node distance, and a positive level of mobility. Boundary cases outside these conditions, such as near-static networks, fully connected networks, and very sparse networks, are discussed in Section V-G.

G. Limitations and Threats to Validity

Although the proposed method achieves high detection accuracy with low overhead in normal operating conditions, several limitations should be considered.

- Observation-time requirement: The method requires a minimum observation period to achieve reliable detection. As shown in Section V-B, the detection accuracy becomes stable only after collecting about 350 neighbor-list snapshots. In short-lived networks, or when nodes frequently join and leave, the algorithm may not collect enough data to confidently identify a Sybil node. The dual-threshold mechanism helps reduce this problem by allowing early suspicion through $threshold\_min$. However, a final de-

cision still requires either reaching $threshold\_max$ or receiving a matching hash from another node.

- Mobility-evasion attacks: An attacker who knows the detection method may try to reduce the mobility similarity between Sybil identities by making their movements less synchronized. However, this strategy also creates a cost for the attacker. It decreases the apparent connectivity of each Sybil identity, reduces the chance of being selected as a routing relay, and limits the benefits gained from using multiple identities. In addition, the sharing of confirmed-Sybil hashes (Section IV-E) reduces the time available for the attacker before detection by another node in the network.
- Performance under sparse or low-mobility conditions: The method depends on changes in neighbor lists to generate useful appearance and disappearance patterns. The experiments in Section V-F show that the method performs poorly when such changes are limited. For example, at very low mobility (0.1–1 m/s), nodes do not move enough to create informative sequences. When the network area is too small compared to the communication range (100×100 m with 100 m range), all nodes remain permanently connected, so neighbor lists do not change. On the other hand, when the communication range is too small compared to the area (10 m in a 500×500 m area), the network becomes very sparse and Sybil identities are rarely observed together. In these situations, the sequences are too short to create a clear difference between $LCS_{baseline}$ and the Sybil-pair LCS, which reduces recall. In such cases.
- Probation period cost: To reduce whitewashing attacks, newly detected identities are placed in a probation period with limited routing privileges, rate limits, and no voting rights, as described in Section IV-E. This creates a usability cost for legitimate new nodes, which must wait until the observation period is completed before fully participating in the network. The duration of this delay depends on $threshold\_min$ and can be adjusted, but removing it completely would weaken the protection mechanism.

VI. Conclusion

In this paper, a lightweight distributed approach for detecting Sybil attacks in MANETs was proposed. Leveraging the node mobility correlation in neighboring nodes, the proposed scheme identifies groups of synchronized identities indicative of Sybil behavior while minimizing computational and communication overhead. The method uses a dual-threshold approach to balance precision and recall. Detection results are securely shared through hash values calculated by the QUARK function, and a probation period is used to mitigate whitewashing attempts. The method's high detection accuracy, demonstrated in simulation results with the INET framework under OMNeT++ and independently confirmed under NS-3, shows that it can achieve over 99.7% accuracy with a negligible false-positive rate, outperforming existing RSSI- and trust-based methods. The proposed method is simple, scalable, and infrastructure-independent, which makes it especially suitable for tactical, search-and-rescue, and IoT-based ad hoc deployments. Our method can operate as a standalone solution or be integrated with existing routing protocols or IDSs. Future extensions will focus on validating the approach in real-world mobile testbeds.

References

[1] J. R. Douceur, "The Sybil attack," in *International Workshop on Peer-to-Peer Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 251–260. [Online]. Available: 10.1007/3-540-45748-8_24

[2] S. Abbas, M. Merabti, D. Llewellyn-Jones, and K. Kifayat, "Lightweight Sybil attack detection in MANETs," *IEEE Systems Journal*, vol. 7, no. 2, pp. 236–248, 2012. [Online]. Available: 10.1109/JSYST.2012.2221912

[3] Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, and X. Zhou, "Multi-channel based Sybil attack detection in vehicular ad hoc networks using RSSI," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 362–375, 2018. [Online]. Available: 10.1109/TMC.2018.2833849

[4] A. Paul and S. Sinha, "Performance analysis of received signal power-based Sybil detection in MANET using spline curve," *International Journal of Mobile Network Design and Innovation*, vol. 7, no. 3-4, pp. 222–232, 2017. [Online]. Available: 10.1504/IJMNDI.2017.089304

[5] S. Vadhana Kumari and B. Paramasivan, "Defense against Sybil attacks and authentication for anonymous location-based routing in MANET," *Wireless Networks*, vol. 23, pp. 715–726, 2017. [Online]. Available: 10.1007/s11276-015-1178-7

[6] Y. Chen, J. Yang, W. Trappe, and R. P. Martin, "Detecting and localizing identity-based attacks in wireless and sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 5, pp. 2418–2434, 2010. [Online]. Available: 10.1109/TVT.2010.2044904

[7] M. Mulla and S. Sambare, "Efficient analysis of lightweight Sybil attack detection scheme in mobile ad hoc networks," in *2015 International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–6. [Online]. Available: 10.1109/PERVASIVE.2015.7086988

[8] S. Abbas, M. Merabti, and D. Llewellyn-Jones, "Signal strength based Sybil attack detection in wireless ad hoc networks," in *2009 Second International Conference on Developments in eSystems Engineering*. IEEE, 2009, pp. 190–195. [Online]. Available: 10.1109/DeSE.2009.27

[9] T. Guven and Z. C. Taysi, "Creating a realistic Sybil attack dataset for inter-vehicle communication," *Peer-to-Peer Networking and Applications*, vol. 18, p. 234, 2025. [Online]. Available: 10.1007/s12083-025-02058-w

[10] R. Almesaeed, "Real-time Sybil attack detection based on channel characterization in VANET," *International Journal of Computer Network and Information Security (IJCNIS)*, vol. 17, no. 4, pp. 71–83, 2025. [Online]. Available: 10.5815/ijcnis.2025.04.05

[11] M. Ayaida, N. Messai, S. Najeh, and K. B. Ndjore, "A macroscopic traffic model-based approach for Sybil attack detection in VANETs," *Ad Hoc Networks*, vol. 90, p. 101845, 2019. [Online]. Available: 10.1016/j.adhoc.2019.01.010

[12] Z. Zhang, Y. Lai, Y. Chen, J. Wei, and Y. Wang, "Detection method to eliminate Sybil attacks in vehicular ad-hoc networks," *Ad Hoc Networks*, vol. 141, p. 103092, 2023. [Online]. Available: 10.1016/j.adhoc.2023.103092

[13] A. Tangpong, G. Kesidis, H. Y. Hsu, and A. Hurson, "Robust Sybil detection for MANETs," in *18th International Conference on Computer Communications and Networks*. IEEE, 2009, pp. 1–6. [Online]. Available: 10.1109/ICCCN.2009.5235387

[14] M. S. Bouassida, G. Guette, M. Shawky, and B. Ducourthial, "Sybil nodes detection based on received signal strength variations within VANET," *International Journal of Network Security*, vol. 9, no. 1, pp. 22–33, 2009. [Online]. Available: 10.6633/IJNS.200907.9(1).05

[15] H. Rajadurai and U. D. Gandhi, "Fuzzy based collaborative verification system for Sybil attack detection in MANET," *Wireless Personal Communications*, vol. 110, no. 4, pp. 2179–2193, 2020. [Online]. Available: 10.1007/s11277-019-06836-7

[16] C. Piro, C. Shields, and B. N. Levine, "Detecting the Sybil attack in mobile ad hoc networks," in *2006 Securecomm and Workshops*. IEEE, 2006, pp. 1–11. [Online]. Available: 10.1109/SECCOMW.2006.359558

[17] N. Khatri, S. Lee, and S. Y. Nam, "Sybil attack-resistant blockchain-based proof-of-location mechanism with privacy protection in VANET," *Sensors*, vol. 24, no. 24, p. 8140, 2024. [Online]. Available: 10.3390/s24248140

[18] M. S. Fallah and M. Mouzarani, "A game-based Sybil-resistant strategy for reputation systems in self-organizing MANETs," *The Computer Journal*, vol. 54, no. 4, pp. 537–548, 2011. [Online]. Available: 10.1093/comjnl/bxq101

[19] S. D. Ubarhande, D. D. Doye, and P. S. Nalwade, "A time stamp-based algorithm to improve security and performance of mobile ad hoc network," *Wireless Networks*, vol. 25, pp. 1867–1874, 2019. [Online]. Available: 10.1007/s11276-017-1640-9

[20] M. Bharti, S. Rani, and P. Singh, "TMTACS: Two-tier multi-trust-based algorithm to countermeasure the Sybil attacks," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 3497–3512, 2022. [Online]. Available: 10.32604/cmc.2022.029072

[21] S. Abbas, M. Merabti, K. Kifayat, and T. Baker, "Thwarting Sybil attackers in reputation-based scheme in mobile ad hoc networks," *KSII Transactions on Internet and Information Systems*, vol. 13, no. 12, pp. 6214–6242, 2019. [Online]. Available: 10.3837/tiis.2019.12.023

[22] A. Bochem, B. Leiding, and D. Hogrefe, "Unchained identities: Putting a price on Sybil nodes in mobile ad hoc networks," in *Security and Privacy in Communication Networks: SecureComm 2018*. Springer International Publishing, 2018, pp. 358–374. [Online]. Available: 10.1007/978-3-030-01701-9_20

[23] J. Hassan, A. Sohail, A. I. Awad, and M. A. Zaka, "LETM-IoT: A lightweight and efficient trust mechanism for Sybil attacks in Internet of Things networks," *Ad Hoc Networks*, vol. 163, p. 103576, 2024. [Online]. Available: 10.1016/j.adhoc.2024.103576

[24] K. Stepien and A. Poniszewska-Maranda, "Security measures with enhanced behavior processing and footprint algorithm against Sybil and bogus attacks in vehicular ad hoc network," *Sensors*, vol. 21, no. 10, p. 3538, 2021. [Online]. Available: 10.3390/s21103538

[25] P. Memarmoshrefi, H. Zhang, and D. Hogrefe, "Social insect-based Sybil attack detection in mobile ad-hoc networks," in *Proceedings of the 8th International Conference on Bioinspired Information and Communications Technologies*, 2014, pp. 141–148. [Online]. Available: 10.4108/icst.bict.2014.258041

[26] S. Mahajan, N. Dahiya, and D. Kumar, "A mechanism of preventing Sybil attack in MANET using bacterial foraging optimization," in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*. IEEE, 2016, pp. 1–5. [Online]. Available: 10.1109/WOCN.2016.7759896

[27] P. V. Rao, K. S. Murthy, V. G. Krishnan, V. Divya, and K. Sathyamoorthy, "Detection of Sybil attack in MANET environment using ANFIS with bloom filter algorithm," *Indian Journal of Computer Science and Engineering*, vol. 13, no. 1, pp. 82–92, 2022. [Online]. Available: 10.21817/indjcse/2022/v13i1/221301058

[28] C. Iwendi, M. Uddin, J. A. Ansere, P. Nkurunziza, J. H. Anajemba, and A. K. Bashir, "On detection of Sybil attack in large-scale VANETs using spidermonkey technique," *IEEE Access*, vol. 6, pp. 47 258–47 267, 2018. [Online]. Available: 10.1109/ACCESS.2018.2864111

[29] K. Suganyadevi, A. Swaminathan, B. Kasi, and M. A. Anju, "Sybil attack detection in VANET using CNN enhanced with chaotic maps and elephant herding optimization for secure data transmission," *Journal of Machine and Computing*, vol. 5, no. 2, pp. 1233–1247, Apr 2025. [Online]. Available: 10.53759/7669/jmc202505097

[30] M. Jamshidi, E. Zangeneh, M. Esnaashari, and M. R. Meybodi, "A lightweight algorithm for detecting mobile Sybil nodes in mobile wireless sensor networks," *Computers & Electrical Engineering*, vol. 64, pp. 220–232, 2017. [Online]. Available: 10.1016/j.compeleceng.2016.12.011

[31] R. Almas-Shehni and K. Faez, "Detection of Sybil nodes in mobile sensor networks using the context of nodes mobility," in *International Symposium on Computer Networks and Distributed Systems*. Springer International Publishing, 2013, pp. 117–128. [Online]. Available: 10.1007/978-3-319-10903-9_10

[32] K. F. Ssu, W. T. Wang, and W. C. Chang, "Detecting Sybil attacks in wireless sensor networks using neighboring information," *Computer Networks*, vol. 53, no. 18, pp. 3042–3056, 2009. [Online]. Available: 10.1016/j.comnet.2009.07.013

[33] J. Grover, M. S. Gaur, V. Laxmi, and N. K. Prajapati, "A Sybil attack detection approach using neighboring vehicles in VANET," in *Proceedings of the 4th International Conference on Security of Information and Networks*, 2011, pp. 151–158. [Online]. Available: 10.1145/2070425.2070450

[34] T. Sharma and L. Singh, "A detection technique for identity based attacks in clustered mobile ad-hoc networks," in *2015 International Conference on Advances in Computer Engineering and Applications*. IEEE, 2015, pp. 893–898. [Online]. Available: 10.1109/ICACEA.2015.7164831

[35] R. Kalaiselvi and P. M. Sundaram, "Machine learning-based strategy to detect Sybil attacks on mobile ad hoc networks," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 16, no. 1, pp. 1227–1238, 2024. [Online]. Available: https://ijcnis.org/index.php/ijcnis/article/view/7038

[36] R. Kalaiselvi and P. M. Sundaram, "Unified framework for Sybil attack detection in mobile ad hoc networks using machine learning approach," *The Scientific Temper*, vol. 16, no. 2, pp. 3774–3782, 2025. [Online]. Available: 10.58414/SCIENTIFICTEMPER.2025.16.2.09

[37] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis & defenses," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, 2004, pp. 259–268. [Online]. Available: 10.1145/984622.984660

[38] G. Manzini, "Suffix trees and suffix arrays," in *The Burrows–Wheeler Transform: Data Compression, Suffix Arrays, and Pattern Matching*. Boston, MA: Springer, 2008. [Online]. Available: 10.1007/978-0-387-78909-5_4

[39] J.-P. Aumasson, L. Henzen, W. Meier, and R. C.-W. Phan, "Quark: A lightweight hash," *Journal of Cryptology*, vol. 26, no. 2, pp. 313–339, 2013. [Online]. Available: 10.1007/s00145-012-9125-6

Samir Selmane received the engineer degree from University of Tebessa, Algeria, in 2010, and received the master degree from University of Bejaia, Algeria, in 2015. Currently, he is a permanent full-time assistant professor at the University of Mila, Algeria, and a Ph.D. student at University of Bejaia, Algeria. He is working on topics related to mobile ad hoc networks and security.

Abderrahmane Baadache received the engineer degree in computer science from the national institute of computing (ESI, ex. INI), Oued Smar, Algiers, Algeria in 2001. The magister and Ph.D degrees from the doctoral school of networking and distributed systems, department of computer science, A. Mira University, Bejaia, Algeria in 2005 and 2012 respectively. His main research topic include wireless networking, security of wireless communication, sensor, mesh and vehicular networks.

Fouzi Semchedine is currently a Full Professor at the University of Setif, Algeria. He received his PH.D in 2011 in computer science from the University of Bejaia, (Algeria). He works in the area of routing, security and quality of service aspects in wireless sensor networks, vehicular networks and hybrid sensor and vehicular networks. He interests also to the localization problem and the compression of data in wireless networks. Actually, he is a head of the research project "Security and Reliability of Wireless Networks (Ad hoc and Sensors)".