International Journal of Computational Methods and Experimental Measurements
https://www.acadlore.com/journals/IJCMEM

A Multi-Phase Framework for Detection and Mitigation of Intentional Packet Dropping Attacks in Mobile Ad Hoc Networks

Polu Srinivasa Reddy, Arshad Ahmad Khan Mohammad*

Department of CSE, GITAM Deemed to be University, 502329 Hyderabad, India

* Correspondence: Arshad Ahmad Khan Mohammad (ibnepathan@gmail.com)

Received: 10-30-2025 **Revised:** 12-11-2025 **Accepted:** 12-24-2025

Citation: P. S. Reddy and A. A. K. Mohammad, "A multi-phase framework for detection and mitigation of intentional packet dropping attacks in mobile ad hoc networks," *Int. J. Comput. Methods Exp. Meas.*, vol. 13, no. 4, pp. 954–970, 2025. https://doi.org/10.56578/ijcmem130414.

Abstract: Mobile ad hoc networks (MANETs) are inherently susceptible to malicious packet-dropping attacks, including black holes, gray holes and selective forwarding attacks that greatly decrease the reliability and performance of MANETs. Current solutions for detecting malicious attacks have large false-positive rates because they often confuse an intended malicious drop with an unintended loss caused by limited resources, traffic congestion and/or the impairment of wireless channels. In addition, current solutions can be vulnerable to acknowledgment (ACK) forgery attacks and consume considerable amounts of energy in continuously monitoring packets. The authors present a comprehensive four-phase framework that synergistically combines route qualification based on available resources, password-based mutual authentication using chaotic map Diffie Hellman password authenticated key exchange (CMDH-PAKE), authenticated digested ACKs based on counters and selectively activating promiscuous monitors to accurately detect and mitigate malicious packet-dropping attacks in MANETs at low power. The authors' solution identifies and excludes honest-but-constrained nodes during route discovery by estimating buffer congestion using exponentially weighted moving average (EWMA) and modeling energy feasibility, thus reducing false positives by up to 73%. Binding keys between cryptographic sessions reduces the potential for ACK forgery and impersonation attacks, and aggregated window-based ACKs reduce energy use by 85%, relative to per-packet ACKs. Selectively activating monitors on demand using only cryptographic evidence of anomalies minimizes the energy used while still maintaining a high level of detection accuracy (above 96%). Simulation results using Network Simulator 3 (NS-3) indicate that the authors' solution has a higher packet delivery ratio (94.2%), shorter end-to-end delay (127 ms), and much lower false-positive rate (3.1%) than other approaches; in addition, the authors' solution uses about 42% less energy than always-on monitoring approaches.

Keywords: Mobile ad hoc networks; Packet dropping attacks; Black hole attack; Gray hole attack; Intrusion detection; PAKE authentication; Resource-aware routing; Promiscuous monitoring; Network security

1 Introduction

Mobile ad hoc networks (MANETs) are an example of an infrastructure-less, wireless communication paradigm, where mobile nodes self-configure to create temporary network topologies without a reliance on either a base station or central controller. Each node is capable of operating as both a terminal device and a router.

The characteristics of MANETs make them appealing to use in such situations as emergency response scenarios, military tactical operations, disaster recovery missions, vehicular ad hoc networks and temporary event networking.

Adaptability and Durability MANETs come up with security weakness; the wireless channel and strong topology with recurring node movement, lack of central monitoring mechanism, limited computational materials and restrained energy finances and inclusive trust in cooperative packet forwarding produces an attack surface that fraudulent nodes can make use of; Intentional packet drop is another attack that deliver severe risk to network reliability. In this type of attack, a compromised node will intentionally discard data packets which it should have forwarded resulting in reduction of packet delivery ratio.

Scope and Methodological Focus: The current study mainly focuses on simulations and analysis through the computational process to determine the quality of the framework presented in the study. In fact, most of the

related literature on MANETs and overall MANETs' security is related to controlled simulations, which have presented repeatable and scalable results. In reality, the computational complexity for the presented framework is linear, depending on the number of nodes participating in the process because the cryptographic calculation is only dependent on the source and destination, and the task of monitoring is only activated when necessary. Although a testbed implementation would have made the study more realistic, the current study is basically simulation-based to ensure repeatable results are obtained.

The sections of related works and security analysis are streamlined to provide clear, focused discussions, with reduced redundancy, emphases on comparative insights among the approaches that exist, and clearer articulation of practical implications of the framework proposed.

1.1 Motivation and Challenges

Several challenges exist in detecting intentional packet dropping attacks in MANETs. First, packet losses in wireless networks arise from multiple sources including buffer overflow due to congestion, medium access collisions, signal attenuation and fading, link breakages due to node mobility, and energy depletion causing nodes to go offline. watchdog-based and acknowledgment-based detection schemes traditionally indiscriminately flag all packet losses as potentially being an attack resulting in excessively high false positive rates where honest nodes that experience temporary resource constraints are wrongly accused and isolated thereby degrading network connectivity and performance.

Second, many of the existing schemes rely on end-to-end or hop-by-hop acknowledgments (ACKs) to verify packet delivery, but these ACKs themselves are vulnerable to forgery by sophisticated attackers. a malicious node can impersonate the destination or intermediate nodes and fabricate false ACKs that will deceive the source into believing the packets were successfully delivered when they were actually dropped. without cryptographic binding between ACKs and legitimate endpoints, detection mechanisms based solely on receipt of ACKs become unreliable.

Third, as mentioned earlier, in terms of resource constraints in a mobile ad hoc network with limited battery power, always-on monitoring will significantly shorten the network's lifetime, and could be considered a denial-of-service attack by nature of the fact that always-on monitoring is inherently wasteful and unnecessary. Furthermore, as stated previously, due to the inherent ambiguity of wireless transmission caused by collisions, the "hidden terminal" problem, and capture effects, there are likely to be cases where the monitoring node cannot determine conclusively whether a packet was intentionally dropped by an attacker or lost due to the normal vagaries of wireless communication.

Fourth, the inconsistency of malicious behavior in gray hole and selective dropping attacks presents unique detection challenges. For example, an attacker might forward all packets of one type normally to avoid detection, while selectively dropping other types based upon source/destination pairs, temporal patterns, etc. This type of intermittent malicious behavior will require sustained monitoring and sophisticated statistical analysis to distinguish from the normal variability of wireless loss rates.

1.2 Contributions

The remainder of this paper addresses these challenges through the following three key contributions:

1. A new route qualification approach for MANETs, which uses energy-aware routing to prevent the routing through genuine but constrained nodes. The proposed method makes use exponentially weighted moving average (EWMA) to calculate buffer congestion and develops energy feasibility to reduce false positives and detach malicious nature from benign losses.

2. Integrating a password enabled mutual authentication using Chaotic Maps Diffie-Hellman Password-Authenticated Key Exchange (CMDH-PAKE) for establishing of cryptographically secure session keys between the source and destination, thereby avoiding impersonation attacks and ACK forgery.

3. Designing counter based authenticated ACKs that accumulate packet delivery confirmations with time, notably reduce ACK overhead while keeping up detection capability with cryptographic integrity verification.

4. Developing selective on-demand monitoring strategy that triggers localized monitoring only when cryptographic evidence advices anomalies, minimizing energy utilization while giving precise localization of malicious nodes.

5. Comprehensive performance evaluations through extensive Network Simulator 3 (NS-3) simulation results demonstrate superior detection accuracy (96.3%), low false positive rates (3.1%), high packet delivery ratios (94.2%) and significant energy savings (42% reduction) compared to existing methods.

This is significant because this effort has woven various techniques together into a comprehensive set. It deals with resource-conscious route establishment, CMDH-PAKE-based ACK messages, windowed loss resilience, and selective promiscuous listening, and they complement each other because they all have gaps that the others fill. Cryptographic verification, statistical loss identification, and behavioral analysis all help and reinforce each other because they understand the role they play together and help mitigate false positives and energy consumption when these methods are used together.

1.3 Paper Organization

The remaining sections of the paper are organized as follows. Section 2 discusses related work on packet dropping attack detection in MANETs and also identifies few limitations of the existing approaches. Section 3 defines the problem statement, network attack models, and the design objectives. Section 4 gives the detailed 4 phase framework that includes resource aware qualification, digested ACKs, mutual authentication, and selective monitoring. Section 5 presents the simulation setup and describes comprehensive performance evaluation results. Section 6 demonstrates formal security analysis giving robustness against various attack vectors. Section 7 compares the proposed framework with modern schemes across multiple metrics. At the end, Section 8 gives the conclusion and includes future research directions.

2 Related Work

MANET packet-dropping-attack detection has been investigated thoroughly, and various studies have developed methods of classification for packet-dropping attack detection as follows; watchdog-based monitoring, acknowledge ment-based verification, reputation systems, and hybrid methods. In this section we provide a critical evaluation of several representative examples of each of the above classifications and identify the problems they pose which serve as motivation for the proposed framework.

Table 1 presents a comparative summary of existing packet dropping detection approaches based on detection rate, false positives, overhead, and ACK forgery resilience.

Table 1. Comparative summary of packet dropping detection approaches

Approach	Detection Rate	False Positives	Over-Head	Acknowledgment (ACK) Forgery Resilience
Watchdog-based schemes	Medium	High	Low	No
ACK-based schemes	Medium–High	Medium	Medium	Partial
Trust-based methods	High	Medium	High	No
Cryptographic ACK schemes	High	Low	High	Yes
Proposed framework	High	Low	Moderate	Yes

2.1 Watchdog and Monitoring-Based Approaches

Marti et al. [1] presented an early watchdog mechanism employing promiscuous monitoring where every node overheard communication from its next-hop neighbors in order to detect packet-forwarding activity. Although the method is relatively simple, it also has many serious deficiencies including: (a) distinguishing between malicious drops and transmission failures of the wireless link, (b) vulnerability to receiver collisions and low transmission power, (c) false misbehavior detection resulting from ambiguous collisions, and (d) continuous monitoring overhead will deplete battery resources quickly. Many subsequent developments attempted to address certain of these deficiencies but did not resolve the fundamental problem of false positives.

Improved watchdog-based monitoring schemes such as 2ACK [2], TWOACK [3], and Enhanced Acknowledgment (EAACK) [4] utilize both overhearing and explicit acknowledgements and cannot distinguish between packet-drops caused by congestion, and those caused by malicious behavior. The EAACK scheme added digital signatures to prevent ACK forgery but resulted in significant cryptographic overhead, and did not consider the resource-constrained nature of honest nodes. Collaborative watchdog-based monitoring schemes [5] utilize cooperative communication among neighboring nodes to reduce ambiguity but result in additional communication overhead, and are vulnerable to collusion attacks.

2.2 Acknowledgment-Based Schemes

End-to-end and/or hop-by-hop ACKs may be used to confirm the successful receipt of a packet in pure acknowledgement-based monitoring schemes. Cooperative Neighbor-based Fault Detection (CONFIDANT) [6] integrated trust management with acknowledgement-based monitoring, however, results in generating numerous control messages. Selective acknowledgment (SACK) mechanisms [7] reduce the number of control messages generated by only acknowledging missing packets, but are still vulnerable to forgery. Cryptographic ACKs utilizing hash-chains or digital signatures [8, 9] prevent forgery, but generate both communication and computation overhead that grows exponentially with network size and traffic load.

Fixed window-based ACK aggregation methods [10] decrease total messages but are static and therefore do not react to different traffic patterns or changing network conditions. Additionally, most ACK aggregation methods rely

on having reliable reverse paths to the sender. In high frequency-changing MANETS this is often not true. Finally, there is no mutual authentication between the source and destination of packets which means they are subject to impersonation attacks where malicious nodes use fake identities to create false ACKs.

2.3 Reputation and Trust-Based Systems

Behavioral reputation systems evaluate nodes based upon how well they forward packets. Collaborative Reputation Mechanism (CORE) [11] utilizes a cooperative reputation system using three types of reputations; subjective reputation (how others see you), indirect reputation (what others think about your neighbors), and functional reputation (your ability to forward packets). Observation-based Cooperation Enforcement in Ad hoc Networks (OCEAN) [12] uses an observation-based cooperation enforcement method however it requires large amounts of memory to store all the reputation information. Bayesian models [13] use probability distributions to represent a nodes behavior however they are limited by their need for a lot of observation data to converge and their inability to quickly respond to rapidly changing topologies.

All of the above methods have inherent problems such as: (a) Bad-mouthing attack and ballot-stuffing attack vulnerabilities where a malicious node will degrade the reputation of other good nodes, (b) gray hole attack vulnerabilities where a malicious node will suddenly start dropping packets, (c) resource constraint vs. malicious node problem where it is difficult to determine if a node has been compromised or just lacks resources, and (d) computational overhead involved in computing and disseminating reputation information. Most importantly, none of the above methods utilize an explicit prevention mechanism for ACK forgery, and therefore rely on majority voting or threshold based decision-making that sophisticated attackers can exploit through coordinated collusion.

2.4 Intrusion Detection Systems

Intrusion Detection Systems (IDS) utilizing machine learning techniques analyze packet streams to detect anomalies indicative of attacks. Techniques including support vector machines (SVM) [14], neural networks [15], ensemble methods [16] have all had some degree of success depending on the application scenario. These methods require large datasets of training examples representing various attack scenarios, are sensitive to changes in network characteristics (concept drift), and cannot provide a low false positive rate without compromising detection capability. Selecting features relevant to the problem domain, classifying packets in real-time remains computationally expensive for resource constrained wireless mobile devices.

Hybrid mechanisms combining specification-based monitoring with anomaly detection [17] improves accuracy but struggle with the challenge of attributing packet. Statistical methods [18] like sequential probability ratio tests provides formal detection which guarantees but require very careful threshold tuning and cannot adapt to different heterogeneous network conditions where different nodes experience differing loss rates due to legitimate factors.

2.5 Cryptographic and Authentication Approaches

The use of cryptographic primitives for the protection of detection mechanisms has recently become an area of research. Hash-based message authentication codes (HMAC) [19] provides integrity of the data but is based on pre-shared keys. Public key infrastructure (PKI) [20] provides strong authentication but the computational cost as well as the communication overhead is too high to be used in MANET environments. Using identity-based cryptography [21] reduces the certificate management burden; however, a trusted third party is required for the generation of keys.

Password authenticated key exchange (PAKE) protocols have been proposed for MANET authentication [22]; however, they have limited application in packet dropping detection. chaotic map-based cryptography [23, 24] can be computed efficiently compared to traditional public key schemes; however, existing MANET security protocols that are based on chaotic maps primarily focus on data confidentiality and integrity rather than attack detection. To our knowledge, there is no prior work on integrating chaotic map-based pake for preventing ACK forgery in packet dropping detection.

2.6 Summary of Limitations

Although many researchers have worked extensively on packet dropping detection schemes, existing schemes have significant limitations:

- High false positive rates due to inability to distinguish malicious drops from resource-constrained or collision-induced losses.
- Vulnerable to ACK forgery and impersonation attacks because there is no binding of session keys.
- High energy consumption due to continuous promiscuous monitoring or per-packet ACK.
- Poor localization accuracy when identifying malicious nodes within multi-hop paths.
- Lack of integrated framework combining resource awareness, cryptographic authentication, efficient acknowledgement and selective monitoring.

Our proposed four phase framework systematically addresses these limitations through synergistic integration of complementary techniques achieving superior detection accuracy, low false positives and low energy consumption.

3 Problem Statement and Design Objectives

By This section defines the network model, attack model, and design objectives that give the gist of the development of our proposed framework.

3.1 Network Model

We are working with a mobile ad hoc network that has N wireless nodes moving around.

• All the nodes use IEEE 802.11 wireless connections with the limited range (100–250 meters).

• Those ranges always are not worth; therefore, network depends upon multi-hop routing. Which means nodes transmit packets to each other so that the messages can travel farther than a single hop.

• Each node has a buffer that handles up to (B_max), and their energy is not unlimited. As they transmit data, their energy (E_res) slowly drains.

• The layout of the network always changes as the nodes are moving. In some situations, they follow random waypoints.

• Routing takes place on-demand with reactive protocols like Dynamic Source Routing (DSR) or ad hoc on-demand distance vector (AODV).

• The wireless channel is not at all perfect. Packet loss varies between 5–15%, due to weak signals, multipath fading, or interference.

• Nodes can switch to indefinite mode to listen to what their neighbors are sending, but it consumes lot of energy.

3.2 Attack Model

We take intentional packet drop attack as a fraction $\alpha\,(0 < \alpha < 1)$ of network nodes that are compromised with the following malicious behaviors:

3.2.1 Black hole attack

Malicious node normally participates in route identification but generally drops everything or almost 90% of the data packets it receives for forwarding creating a Black Hole, where packets vanish thereby minimizing packet delivery ratio.

3.2.2 Gray hole attack

One of the node acts as a malicious node which is not actually true always. Sometimes it forwards packets like it should. Other occasions, it selectively drops packets, either based on probability or certain traffic. This can become tough to catch it since it does not always act malicious.

3.2.3 Selective forwarding attack

When this node drops packets, it aims specific destinations, forwarding the remaining as they are. This attack type can cut the important nodes, while the network feels it as innocent.

3.2.4 Attack sophistication

We consider that the attacker has the following capabilities:

• Can mimic other nodes by forging source addresses in ACK messages.

• May suppress or delay forwarding ACKs from downstream nodes.

• Can collude with other malicious nodes to coordinate attack strategies.

• Cannot break cryptographic primitives (pre-shared passwords, hash functions).

• Have awareness about network protocols rather than just cryptographic keys.

3.3 Design Objectives

Our framework aims to tackle the below issues:

• High detection accuracy: Identify the malicious nodes with at least 95% detection rate.

• Low false positives: Keep false positives at 5% or lower.

• Avoid forged ACKs: Assure that ACKs are tied to real source-destination so that the attackers can't fake them.

• Energy efficient: Minimize the monitoring overhead. It must not use more than 10% energy compared to normal.

• Pinpoint the problem: Identify accurately which node is being malicious in any route so that we can fix the things fast.

• Scalability: Maintain detection performance as network size increases up to 100 nodes with varying mobility patterns.

• Low Communication Overhead: Reduce control message traffic compared to per-packet ACK schemes by $\geq$80%.

• Robustness: Remain effective under varying attack intensities (α = 10–40%), mobility speeds, and network densities.

4 Proposed Work

The proposed solution is an integrated and multi-step process whereby every step confirms and strengthens one another. Through route qualification with awareness, the uncertainty related to constrained nodes can be removed. Authenticated acks provide credible proof of losses, and the process to check all routes gets triggered only if anomalies persist.

Resilient We combined a framework to catch and counter Intentional Packet Dropping (IPD) attacks in MANETs. It in 4 connected phases, each doing its own task but working together: Firstly, the routes are qualified based on resources. Nodes check their own energy levels and buffer congestion before joining a route. If a node is running low or overloaded on power, it leaves. This simple step will reduce false alarms from genuine nodes dropping packets. Next, we use a password enabled mutual authentication approach CMDH-PAKE. Before transmitting the data, both source and destination nodes prove to each other they are legitimate, using a pre-approved password to generate a session key. This ensures that the ACKs cannot be faked. If the authentication itself is failed, then the session will not start.

4.1 End-to-End Operational Idea

For data flow from source (S) to destination (D), the system identifies IPD through a four-staged process that combines (1) proactive filtering, (2) secure authentication, (3) efficient ACK, and (4) on-demand monitoring:

• At the time of route discovery, nodes estimate the buffer congestion and energy levels.

• After the route establishment, S and D mutually authenticate with a pre-approved password to derive the session key K_SD. This avoids impersonation of D and assures ACKs are not faked.

• Instead of sending a ACK for each and every packet, the destination node collects receipts after a certain time or after a set number of packets and then sends 1 ACK back to the source.

• If ACK is not received due to time out or doesn't check out, the system ends up monitoring.

This entire process is layered to conserve energy: step-1 happens while setting up the route, step-2 is a single time check per session, step-3 runs in batches periodically, and step-4 only when there is some trouble.

4.2 Phase-I: Resource-Aware Route Qualification

For this, we make use of 2 checks: First model is used to calculate buffer congestion using an EWMA, and the other model is used for energy feasibility. Both models help us to decide if a node is good for the route or not.

4.2.1 Buffer Congestion Estimation

Instantaneous queue length Q_inst fluctuates due to bursty traffic; EWMA provides a stable estimate:

$$Q\_avg(t) = \alpha Q\_inst(t) + (1 - \alpha)Q\_avg(t - 1) \tag{1}$$

where, α (typically 0.1–0.3) is chosen based on traffic patterns—lower for stable environments. A small value smooths transients, preventing overreaction to short bursts while capturing sustained congestion.

Threshold:

$$Q\_th = 0.75 \times B\_max \tag{2}$$

where, B_max is maximum buffer size. The 75% threshold preserves 25% headroom for bursts and control packets, mitigating overflow drops. If $Q\_avg \geq Q\_th$, the node is prone to unintentional drops and is excluded.

4.2.2 Energy Feasibility Modeling

Packet forwarding incurs energy costs:

$$E\_pkt = E\_rx + E\_proc + E\_tx \tag{3}$$

where, E_rx, E_proc, E_tx represent receive, process, and transmit energy respectively.

Sustainability ratio:

$$E\_s = E\_res\ /E\_pkt \tag{4}$$

Eligibility threshold:

$$E\_th = (0.75 \times E\_init)/E\_pkt \tag{5}$$

Nodes below E_th are "honest but weak" and likely to drop packets unintentionally; excluding them prevents false accusations. The 75% factor ensures sustainability for multi-hop routes, chosen empirically to balance availability and reliability.

4.2.3 Composite eligibility criteria

A node n_j is eligible if:

$$(Q\_avg\,(n\_j) < Q\_th) \wedge (E\_s\,(n\_j) > E\_th) \tag{6}$$

This ensures both resources are adequate. We extend AODV routing by adding fields to route request (RREQ): Q_avg, E_res (or compressed status bits). Each intermediate node evaluates eligibility before forwarding RREQ. Ineligible nodes silently drop RREQ, naturally forming resource-aware paths.

Notation explanation (free style) to keep things clear, here's a quick recap of the main symbols used in the proposed algorithms. R_j is the packet forwarding rate we observe at node j. B_j stands for the node's available buffer capacity. E_j represents the residual energy of node j. The EWMA parameter α sets how much smoothing we apply when estimating congestion, and β is the packet loss tolerance used during ACK verification. The monitoring threshold θ marks the boundary for deciding whether behavior is malicious. All other variables follow the standard MANET routing.

Algorithm 1: Resource-aware qualification.

Input:

$Q_inst, Q_avg_prev, B_$max,$E_$res , $E_init, E_$rx , E_proc, E_tx, α

Output: Eligible (true/false)

1 $Q\_avg = \alpha * Q\_inst + (1 - \alpha) * Q\_avg\_prev$

2 $Q\_th = 0.75 * B\_max$

3 if $Q\_avg\ = Q\_th$ then return FALSE

4 $E\_pkt = E\_rx + E\_proc + E\_tx$

5 $E\_th = (0.75 * E\_init)/E\_pkt$

6 $E\_s = E\_res \div E\_pkt$

7 if E, $s <= E\_th$ then return FALSE

8 return TRUE

This phase preemptively eliminates the primary source of unintentional losses, sharpening the focus on intentional threats.

Routing Protocol Generalizability: The existing implementation integrates resource-aware qualification with AODV by extending RREQ messages; however, the proposed framework is not intrinsically bound with AODV. In source-routing protocols such as DSR, the very same eligibility checks can be performed during route reply elaboration, while proactive protocols such as Optimized Link State Routing (OLSR) periodically disseminate resource status by control messages or embed it into link-state metrics. As the framework does not depend on specific routing decisions and requires only local resource appraisal, authenticated ACKs, and selective monitoring, it may easily adapt to other MANET routing protocols by making minor protocol-specific modifications.

4.3 Phase-II: Password-Based Mutual Authentication and Session Key

Without cryptographic confirmation of D's identity, attackers can impersonate D (faking ACKs), forge ACKs, or manipulate monitoring triggers. PAKE ensures the password is never transmitted, binding the session key to prevent Man-in-the-Middle (MITM) attacks.

4.3.1 Chebyshev polynomial-based key exchange

Leveraging Chebyshev polynomials for enhanced security over traditional Diffie-Hellman:

Recurrence:

$$T\_k(x) = 2xT\_\{k-1\}(x) - T\_\{k-2\}(x)\, mod\, N \tag{7}$$

Composition:

$$T\_m\,(T\_n(x)) = T\_\{mn\}(x) \tag{8}$$

This provides semi-group properties resistant to discrete logarithm attacks. Both parties generate ephemeral secrets a and b, exchange commitments A and B, compute shared K^*, and derive K_SD bound to password PW_SD. Explicit tokens confirm mutual knowledge.

4.3.2 Protocol steps

1. Ephemeral exchange:

$$S \rightarrow D : (ID\_S, A = T\_a(x)\, mod\, N) \tag{9}$$

$$D \rightarrow S : (ID\_D, B = T\_b(x)\, mod\, N) \tag{10}$$

2. Shared secret:

$$K^* = T\_a(B) = T\_b(A) = T\_\{ab\}(x)\, mod\, N \tag{11}$$

3. Password-bound session key:

$$K\_SD = H\left(K^* \| ID\_S \| ID\_D \| PW\_SD\right) \tag{12}$$

where, H () is a cryptographic hash function (e.g. SHA-256).

4. Authentication tokens:

$$Auth\_S = H\left(K\_SD \| \text{“ } S \text{ ”}\right) \tag{13}$$

$$Auth\_D = H\left(K\_SD \| \text{“ } D \text{ ”}\right) \tag{14}$$

where, D verifies $Auth\_S$ to accept S, proving S's knowledge of K_SD. S verifies $Auth\_D$ similarly. MITM fails without PW_SD, as relaying A/B yields incorrect K_SD. Parameters: N is a large prime, x is public; chosen for computational security (e.g., 1024-bit N). S initiates; both compute locally and exchange. Triggers post-route establishment, before data flow. Ephemerals are random (e.g., 256-bit); PW_SD is pre-shared. This ensures ACKs are verifiable only by legitimate S/D, blocking forgery.

Algorithm 2: Password-based mutual authentication (CMDH-PAKE style).
Public: (N,x), Hash $H()$
Secret: PW_SD
Output: Session key K_SD or Abnormal Termination (ABORT)
Source S:
1 choose random a
2 $A = T\_a(x) \, mod \, N$
3 send (ID_S, A) to D
4 receive (ID_D, B)
5 $K^* = T\_a(B) \, mod \, N$
6 $K\_SD = H(K^* || ID\_S || ID\_D || PW\_SD)$
7 $Auth\_S = H(K\_SD || \text{“}S\text{”})$; send $Auth\_S$
8 receive $Auth\_D$; verify $Auth\_D == H(K\_SD$——“D”)
9 if fail - $>$ ABORT else ACCEPT K_SD
Destination D:
1 receive (ID_S, A)
2 choose random b
3 $B = T\_b(x) \, mod \, N$
4 send (ID_D, B)
5 $K^* = T\_b(A) \, mod \, N$
6 $K\_SD = H(K^* || ID\_S || ID\_D || PW\_SD)$
7 receive $Auth\_S$; verify $Auth\_S == H(K\_SD || \text{“}S\text{”})$
8 if fail - $>$ ABORT
9 $Auth\_D = H(K\_SD || \text{“}D\text{”})$; send $Auth\_D$

Password Establishment and Management: The proposed CMDH-PAKE mechanism assumes the lightweight pre-shared password (PW_SD) between communicating source–destination pairs. Such assumptions are realistic for many MANET scenarios, such as military units, emergency response teams, and vehicular clusters, where nodes are provisioned during deployment or share credentials by way of prior secure contact. Password update or revocation can be done easily by invalidating the corresponding session and reinitiating authentication with a new password upon route change or suspected compromise. Password exposure and replay risks are thus minimized due to the fact that it is never transmitted over the network and session keys are derived using fresh ephemeral values. The underlying password distribution methodology itself keeps the framework independent, which allows seamless integration of most existing MANET key pre-distribution or out-of-band credential mechanisms.

4.4 Phase-III: Counter-Based Authenticated Digested ACK

Per-packet ACKs impose high overhead (routing, congestion, energy) and vulnerability to forgery/suppression. Aggregated window-based ACKs reduce this while maintaining detection capability.

4.4.1 Window configuration

Window size T_s (time- or packet-based, e.g., 10s or 50 packets).

$$ACK\ wait : T\_ack = T\_s + RTT + \delta \tag{15}$$

where, δ is a margin for mobility/Medium Access Control (MAC) delays (e.g., 1–2 s).

4.4.2 ACK structure and integrity

$$ACK = \langle SID, DID, WID, T\_s, C\_rx, Seq\_s, Seq\_e \rangle \tag{16}$$

where, C_rx is the received count, and sequence range (Seq_s, Seq_e) prevents replays.

$$Integrity : d = H\left(ACK \| K\_SD\right) \tag{17}$$

$$Verification : H\left(ACK \| K\_SD\right)? = d \tag{18}$$

4.4.3 Detection logic

Timeout or digest mismatch → SUSPECT (triggers Phase-IV)
Add tolerance:

$$\text{If } C\_rx \geq (1-\beta) C\_tx \text{ then OK; else SUSPECT}$$

where, β = 5–10% (tuned for channel loss). This parameter absorbs normal wireless losses, preventing unnecessary monitoring. S sends data, waits/verifies; D counts, builds/sends ACK via reverse path. Triggers per window during transmission. T_s balances overhead/detection latency; β is derived from empirical BER. The mechanism detects drops/suppression cryptographically; forgery fails without K_SD.

Distinguishing between mobility-induced loss and malicious dropping: In highly mobile MANETs, packet or ACK loss may be due to transient route breaks rather than malicious behavior. The proposed framework dispels this ambiguity via three complementary mechanisms. First, the ACK timeout T_ack explicitly incorporates round-trip time and an additional mobility margin δ, allowing sufficient time for route repair before suspicion is raised. Second, the loss tolerance parameter β absorbs moderate packet losses due to wireless impairments or mobility, thus preventing premature suspicion. Finally, no node is classified as malicious based solely on ACK loss; instead, ACK anomalies serve only to trigger Phase-IV selective monitoring, where localized promiscuous observation confirms intentional dropping. This two-stage decision process significantly reduces false positives due to legitimate mobility induced losses.

Algorithm 3: Counter-based authenticated digested ACK:
Input: T_s, RTT, δ, K_SD
Output: OK or SUSPECT
Source S:
1 start window WID
2 send DATA packets for duration T_s (or N_w packets)
3 wait until T_ack = T_s + RTT + δ
4 if no ACK received -> return SUSPECT
5 compute $d' = H(ACK||K\_SD)$
6 if $d' != d$ - > return SUSPECT
7 if $C\_rx < (1-\beta) * C\_tx$ - > return SUSPECT
8 return OK
Destination D:
1 during T_s, count received packets C_rx and seq range ($Seq\_s ... Seq\_e$)
2 build ACK = ⟨SID, DID, WID, T_s, C_rx, Seq_s, Seq_e⟩
3 $d = H(ACK||K\_SD)$
4 unicast (ACK, d) to S via reverse path

4.5 Phase-IV: Selective Promiscuous Monitoring

Always-on monitoring drains energy and yields false positives from collisions/ambiguous overhearing. We activate it only on SUSPECT, localizing to hop-pairs.

4.5.1 Monitoring architecture

On route $S \rightarrow n\_1 \rightarrow n\_2 \rightarrow ... \rightarrow n\_k \rightarrow D$:

S monitors n_1, n_1 monitors n_2, etc. Each compares relayed packets R_j (to next) vs. overheard forwards F_j (by next).

$$Dropcount : \Delta\_j = R\_j - F\_j \tag{19}$$

$$Decision : \Delta\_j \geq \theta \rightarrow malicious \tag{20}$$

$$Threshold : \theta = \alpha R\_j + \gamma \tag{21}$$

where, α is the expected loss rate (e.g., 5%) and γ is a margin (e.g., 2–5 packets). This accounts for normal losses and is robust against transients.

4.5.2 Isolation and recovery

Broadcast control alert:

$$CTRL = \langle type, ID\_mon, ID\_mal, WID, time, TTL \rangle \tag{22}$$

where, TTL limits flooding. Nodes update blacklists and invalidate routes. Packets are matched via ($flow\_id$, seq_no) or header hash $h = H(seq||flow||K\_SD)$, avoiding flow confusion. Each route node monitors its successor upon trigger. Triggers on Phase-III SUSPECT; runs for interval T_m (e.g., 5–10 s). θ is derived from simulation; T_m enables timely detection. This localizes drops precisely, enabling isolation without global overhead.

Algorithm 4: On-demand promiscuous monitoring & localization:
Input: θ, T_m (monitor interval), K_SD-bound SUSPECT trigger
Output: Malicious node ID or NONE
Triggered when Phase-III returns SUSPECT:
For each node i on the current route:
1 set monitor mode for next hop j
2 $R\_j = 0$, $F\_j = 0$
On each forwarded packet p from i -> j:
3 R_j++
On overhearing j forwarding same packet p (matching seq/hash):
4 F_j++
Every T_m:
5 $\Delta = R\_j – F\_j$
6 if Δ >= θ:
7 broadcast CTRL $\langle i, j$, WID, time, TTL$\rangle$
8 blacklist j; initiate route repair excluding j
9 return j
10 else continue monitoring

4.5.3 Mitigation of intentional packet dropping

The framework effectively mitigates different attack variants:

Black Hole Attack:

Black hole drops most packets $\rightarrow$ C_rx is low or ACK is absent $\rightarrow$ SUSPECT triggered $\rightarrow$ detects high Δ at malicious hop $\rightarrow$ node blacklisted; routes repaired.

Gray Hole/Selective Dropping:

Intermittent drops $\rightarrow$ degraded C_rx beyond β $\rightarrow$ monitoring triggers $\rightarrow$ cumulative $\Delta \geq \theta$ flags node over time.

ACK Forgery/Suppression:

Forgery fails (digest mismatch without K_SD); suppression causes timeout $\rightarrow$ monitoring localizes suppressor.

False Accusation Prevention:

Constrained nodes excluded; β and θ tolerate wireless losses, minimizing errors. This ensures robust, low-overhead mitigation, with cryptography handling forgery and monitoring addressing localization.

4.6 Combined Master Algorithm

Input: Data flow $S \rightarrow D$, PW_SD, thresholds (Q_th, E_th, β, θ), window T_s
Output: Reliable delivery + malicious isolation
1 Route Discovery (AODV):
2 Accept only nodes passing Resource Aware Check (Alg-1)
3 Run Mutual Auth Key Agreement (S, D, PW_SD) -> K_SD (Alg-2)
4 For each window T_s:
5 Send packets; run Authenticated Window ACK (Alg-3)
6 if OK: continue
7 else:
8 Run Selective Promiscuous Monitoring (Alg-4)
9 Isolate malicious node; reroute excluding it
10 Resume transmission

5 Performance Evaluation

Here, we get into a detailed performance check of the framework, working with NS-3 tool 3.36 version. We observed on how well the tool detects threats, how many false alarms show up, how packets get delivered reliably, and how much time it took for data to travel across the network. Above all, we tracked energy use and any additional overhead. We tested this under different attacks, changing mobility, and a variety of network sizes.

Simulation Scope and Limitations: To compare performance, we can fall back on typical mobility and traffic simulations to ensure comparability with current MANET security literature. While they are good representatives of real behavior, actual conditions may deviate, depending on capabilities, irregular movements, and/or interference in real-world settings, which would otherwise be taken into account in absolute performance measurements. In any case, the performance gains observed over a wide spectrum of scenarios are not affected by such differences. In the future, a natural path would be to conduct larger-scale simulations or actual deployments.

Energy Overhead Analysis: In simple words, analysis reveals that the energy consumed over and above in our scheme is primarily because of the CMDH-PAKE authentication and selective and targeted promiscuous mode monitoring. In the CMDH-PAKE phase, only the communicating nodes are involved and not the routers, thus keeping the cryptographic costs light. In the case of the second phase, the energy used for monitoring is minimized and restricted to small regions of the network, and it is triggered only in case there are anomalies in the authenticated acknowledgement messages, not otherwise. Our simulations have found that the cumulative energy overhead is quite reasonable and, in fact, insignificant compared to the enhancement in the detection accuracy and the delivery ratio achieved through our scheme.

5.1 Simulation Setup

The simulation environment is set up with following parameters:

- Network area: 1000 m × 1000 m;
- Number of nodes: 25, 50, 75, 100;
- Transmission range: 250 meters;
- Mobility model: Random waypoint with pause time 0–10 s;
- Node speed: 0–20 m/s (pedestrian to vehicular);
- Traffic: CBR (Constant Bit Rate) over UDP, 512 byte packets;
- Traffic load: 5, 10, 15, 20 packets/second;
- Malicious node percentage (α): 10%, 20%, 30%, 40%;
- Attack types: Black hole (100% drop), Gray hole(50% drop);
- Simulation time: 300 s;
- MAC protocol: IEEE 802.11b DCF;
- Routing protocol: AODV with resource-aware extensions;
- Initial energy: 100 J per node;
- Tx power: 0.660 W, Rx power: 0.395 W;
- Window size (T_s): 10 s/50 packets;
- Wireless loss tolerance (β): 8%;
- Monitoring threshold (θ): 0.05 × R_j + 3.

5.2 Detection Accuracy and False Positives

Detection accuracy proves on how well the system spotted the malicious nodes. The false positives tell us how often it wrongly alarmed the genuine ones. When the attack intensity is in between 10–40%, the framework manages to obtain an average detection accuracy of 96.3% for black-hole attacks, and 94.7% accuracy for gray-hole attacks. Even in difficult times, the false positive does not go up, it is steady at 3.1%. This proves that the system does a good job apart from harmless glitches and real threats.

Resource-aware qualification weeds about 78% of honest nodes in Phase-I. Cryptographic authentication blocks each ACK forgery attempt in Phase-II of our tests, nothing is missed. Selective monitoring zeroes in Phase-IV on the exact malicious hop in a multi-hop path with 97.4% accuracy.

5.3 Packet Delivery Ratio

Packet Delivery Ratio (PDR) basically shows the number of data packets reach to the destination. In a regular network, you will get PDR about 95.3%. But if 40% of the nodes turn into black-holes, PDR drops very fast, then the PDR go down to 47.2%. Now, with the proposed framework, things got better. Even when 40% of the nodes drop, it maintains PDR high at 94.2%. It is because; the proposed approach kicks out the bad actors (approximately in 12.3 seconds) and finds optimal paths for the data. Even if 40% of the nodes turn into gray-holes, the system maintains PDR at 91.8%. So, it is evident that against attackers it is always predictable.

5.4 End-to-End Delay

It covers transmission, queuing, propagation and processing time. With this new framework, the extra delay is very much minimal. The average end-to-end delay rises from 118 ms to 127 ms with regular AODV when there are 20% attacks. The slight increase happens because the system repairs the routes and sometimes picks up lengthy paths to dodge bad nodes. Unlike always monitoring that pushes the delays past 400 ms, this framework always keeps things running smoothly.

5.5 Energy Consumption

Energy efficiency is really important for mobile nodes that are powered by battery. We traced on energy utilized by each node while sending, receiving, or idle. Our framework uses about 8.7% more energy than the basic AODV without detection, just because of authentication and selective monitoring when required. It still saves a lot of power. It reduces energy consumption by 42% on watchdog schemes (which has 15.2% more), and by 68% compared to setups that sends ACK for every single packet (which has 27.4% more).

If you split the energy usage, authentication takes only 1.8%, selective monitoring accounts for 4.6%, and processing the ACKs uses 2.3%. The on-demand monitoring technique is really best. In many occasions, monitoring only consumes about 14.2% of communication sessions. Even if attacks go up to 40%, monitoring triggers in 31.7% of sessions. So, despite of draining the battery, you get strong protection and a smart balance between energy use and security.

Energy Model Considerations: In the given energy model, the focus is on those components of MANET protocols which are responsible for energy consumption, such as transmission, reception, encryption, and monitoring. The reason for not including the energy for idle listening, mode switching, and retransmission is to have a relatively lower absolute energy, although the trend will remain the same because both perspectives are affected equally. Thus, the energy consumption analysis is presented conservatively in the results.

5.6 Communication Overhead

Communication overhead is used to trace about the network's traffic sending control messages. Using window-based ACKs in Phase-III reduces ACK messages by 85%. So, if you transmit 1000 data packets, you get only 20 ACKs rather than 1000 separate ones. When you consolidate all the control messages, combined ACKs, authentication checks, and isolation alerts, they sum up to 6.3% of data traffic during a 20% attack. If the attack goes up to 40%, overhead increases to 11.2% as the system requires more detection and rerouting.

5.7 Scalability Analysis

We verified the scalability of system from 25 to 100 nodes. Detection accuracy changed was very negligible; it was steady all the time, dropping very little from 96.1% to 95.8% as the network grows. This confirms that the localized monitoring approach is good for scaling. The false positive rate increases a little, from 2.8% to 3.4%, just because more routes mean more chance to go wrong during indefinite monitoring. PDR reduces little from 95.1% to 93.4% as more nodes were added to the network. It is because the average path between nodes increases. On the positive side, each node's energy is almost the same, proving that the selective monitoring approach incurs overhead as the network grows.

5.8 Impact of Mobility

As nodes move from idle to zipping around at 20 ms, the shape of the network changes. This will affect the route stability and can incur problems. Detection accuracy goes down a bit from 97.1% when things are at normal speed to 94.9% at top speed. When nodes accelerate, routes can break even before monitoring. But, the framework addresses this well and it reacts fast. The re-authentication is done in less than 230 ms whenever route changes. Furthermore, it adapts the monitoring period by 20% during high mobility, assuring there is enough time to catch issues. False positives slightly increase from 2.6% to 3.8% as mobility increases. It is just because the system sometimes considers normal connection drops as monitoring windows shortage.

5.9 Sensitivity Analysis of Key Parameters

We tested the resilience of the framework by tweaking a few key knobs: α for EWMA smoothing, β which dictates how much loss we are willing to tolerate, and θ, which controls the threshold for monitoring. The system remains stable as long as α is within the range of 0.1–0.3, but making α any higher renders it sensitive to short lived traffic spikes. Similarly, maintaining β within 5–10% provides an adequate trade-off between accurate detection while inherently absorbing losses due to mobility without dampening an attack alarm. For θ, choosing moderate thresholds provides a reasonable trade-off that offers quick detection without many false alarms. In summary, the framework shows robustness with the variation of the parameters within a reasonable range.

5.10 Detection Latency Analysis

It is not only important to be accurate in detection but also to be able to react quickly to evasive tactics such as that of gray-hole attacks. Analysis has shown that malicious nodes are detected with certainty only after few observation cycles of ACK anomalies occur, averaging approximately a few seconds in moderately active networks when there is conclusive proof of irregular packet loss.

6 Security Analysis

This section provides security analysis demonstrating the robustness of framework against various attacks including impersonation, ACK forgery, replay attacks, collusion scenarios, and man-in-the-middle attacks.

6.1 ACK Forgery Prevention

Theorem 1: An adversary without knowledge of the session key K_SD cannot forge a valid ACK that will be accepted by the source S.

Proof:

The ACK verification at source S requires computing $d' = H(ACK||K\_SD)$ and verifying $d' = d$. For an adversary to forge ACK, they must produce (ACK, d) such that $d = H(ACK||K\_SD)$ without knowledge of K_SD. Given the cryptographic collision resistance and pre-image resistance of hash function H (e.g., SHA-256 with 2^{256} output space), finding d for arbitrary ACK content without K_SD requires 2^{256} hash computations on average, which is computationally infeasible. Therefore, forged ACKs will fail verification with probability negligibly close to 1.

6.2 Impersonation Attack Resistance

Theorem 2: The CMDH-PAKE authentication protocol prevents adversaries from impersonating legitimate source or destination nodes without knowledge of the pre-shared password PW_SD.

Proof:

During authentication, both parties exchange commitments $A = T\_a(x) \, mod \, N$ and $B = T\_b(x) \, mod \, N$, then compute $K^* = T\_ab(x)$ and derive $K\_SD = H\left(K^* \| ID\_S \| \, ID\_D \| PW\_SD\right)$. An adversary attempting impersonation must produce valid authentication tokens $Auth\_S = H\left(K\_SD \| \text{“}S\text{”}\right)$ or $Auth\_D = H(K\_SD \| \text{“}D\text{”})$. Without PW_SD, the adversary cannot compute correct K_SD even with intercepted A and B, because K_SD depends on K^* and PW_SD. The security of Chebyshev polynomial-based key exchange reduces to the discrete logarithm problem over chaotic maps, which is computationally hard for appropriately chosen parameters ($N \geq 1024$ bits). Therefore, impersonation succeeds only with negligible probability $1/2\text{^}\mathrm{k}$ where k is the security parameter.

6.3 Man-in-the-Middle Attack Prevention

Theorem 3: An active adversary positioned between source S and destination D cannot successfully execute a man-in-the-middle attack to compromise the session key.

Proof:

Consider adversary M intercepting and relaying authentication messages. M receives A from S and B from D, and can replace these with M's own commitments $A' = T\_m(x)$ and $B' = T\_n(x)$. This creates two separate sessions: $S \leftrightarrow M$ with shared secret $K^*\_SM = T\_am(x)$, and $M \leftrightarrow D$ with shared secret $K^*\_MD = T\_bn(x)$. However, when deriving session keys, S computes $K\_SD = H(K^*\_SM \, \| ID\_S \| \, ID\_D \| PW\_SD \,)$ while M can only compute $K\_SM = H\left(K^*\_SM \, \| ID\_S \| \, ID\_M \| PW\_SD\right)$ or similar variants using different identity combinations. The hash binding to ID_S, ID_D ensures M cannot produce authentication tokens that satisfy both S 's verification (expecting $Auth\_D$ from true ID_D) and D's verification (expecting $Auth\_S$ from true ID_S) simultaneously. Therefore, authentication fails and the MITM attack is detected.

6.4 Replay Attack Resistance

Theorem 4: The framework prevents replay attacks where adversaries retransmit previously valid ACKs or authentication messages.

Proof:

ACK replay: Each ACK includes window ID and sequence number range (Seq_s, Seq_e) that change with each window. Source S maintains state of expected WID and sequence numbers. Replaying old ACK with outdated WID/sequence will be rejected as stale. Additionally, each session uses unique K_SD derived from fresh ephemeral secrets (a, b), so ACKs from previous sessions have different digest values and fail verification. Authentication replay: Authentication tokens $Auth\_S$ and $Auth\_D$ are session-specific through K_SD. Even if intercepted and replayed, they correspond to specific ephemeral exchanges (A, B) that legitimate parties will not repeat (fresh randomness ensures $A \neq A'$ and $B \neq B'$ across sessions with overwhelming probability). Therefore, replayed authentication fails due to mismatched ephemeral values.

6.5 Collusion Attack Analysis

Collusion scenarios involve multiple malicious nodes coordinating their behavior to evade detection. We analyze two primary collusion strategies:

Cooperative Forwarding Collusion: Malicious nodes M1 and M2 on the same route may attempt to conceal drops where M1 drops packets but M2 generates fake transmissions to deceive M1's monitor. However, packet matching via cryptographic hash $h = H(seq \| flow \| K\_SD)$ prevents M2 from generating validlooking forwards without the actual packets. M2 cannot compute correct hashes without K_SD, so fake transmissions are detected as hash mismatches. Simulation results show collusion reduces detection accuracy by only 2.3% (from 96.3% to 94.0% with 3 colluding malicious nodes), demonstrating resilience.

ACK Suppression Collusion: Multiple malicious nodes may coordinate to suppress ACK forwarding along the reverse path. However, ACK timeout ($T\_ack = T\_s + RTT + \delta$) triggers monitoring regardless of which specific hop suppresses ACKs. Selective monitoring then localizes the responsible malicious node through hop-byhop observation. Even with 40% malicious nodes attempting coordinated suppression, localization accuracy remains 95.2%.

6.6 Privacy Considerations

The framework maintains communication privacy through session key protection. Password PW_SD is never transmitted, preventing password exposure to eavesdroppers. Authentication tokens $Auth\_S$ and $Auth\_D$ do not reveal PW_SD due to hash pre-image resistance. Promiscuous monitoring observes packet headers for counting but does not decrypt packet payloads (payload encryption using K_SD remains orthogonal to the detection mechanism and can be added). Control messages CTRL broadcast malicious node identities but do not reveal traffic content, flow patterns, or communication endpoints beyond what is inherently observable in unencrypted routing headers.

Practical takeaway: The above result ensures that, within the proposed framework, middlemen cannot forge or replay ACK messages. Thus, any identified anomaly in ACKs reliably flags abnormal packet forwarding behaviour, not a cryptographic tampering.

7 Comparative Analysis

This section compares the proposed framework against state-of-the-art packet dropping detection schemes across multiple performance and security metrics.

7.1 Comparison with Watchdog-Based Schemes

The basic watchdog mechanism [1] achieves 89.3% detection accuracy but suffers from 24.7% false positive rate due to inability to distinguish malicious drops from wireless collisions and congestion. Our framework reduces false positives to 3.1% (87.4% improvement) through resource-aware route qualification that preemptively excludes honest-but-constrained nodes. Energy consumption comparison shows Watchdog requires continuous monitoring overhead of 15.2% compared to our selective approach at 8.7% (42.8% reduction). Collaborative watchdog [5] improves detection to 92.1% but maintains high false positives (18.4%) and introduces 23.1% communication overhead from neighbor coordination messages, compared to our 6.3% overhead.

7.2 Comparison with Acknowledgment-Based Schemes

TWOACK [3] provides 93.4% detection accuracy with 7.2% false positives but generates per-hop ACKs creating 31.4% communication overhead. Our aggregated window-based approach achieves higher detection (96.3%) with 85% less ACK traffic. EAACK [4] includes digital signatures for ACK integrity instead incurs 27.4% energy overload from Rivest–Shamir–Adleman (RSA) signatures compared to our hash-based verification at 8.7% penalty. Additionally, EAACK has 12.1% false positive rate as it deficits resource-aware qualification, while our framework attains 3.1%.

7.3 Comparison with Acknowledgment-Based Schemes

CONFIDANT [6] will keep up reputation scores attaining 90.8% detection but need 42–67 seconds detection time because of reputation convergence delay, in comparison to our 12.3 seconds. False positive rate of 16.3% results from slow adaptation to legitimate resource constraints, in contrast to our proactive filtering at 3.1%. OCEAN [12] gives 91.7% detection with 14.8% false positives but utilizes significant memory (on an average of 2.4 MB per node) for reputation tables in a 100-node network, while our stateless cryptographic approach needed minimal storage ($<$1 KB per active session).

7.4 Comparison with Machine Learning Approaches

SVM-based intrusion detection [14] obtains 94.2% detection accuracy with an excessive training (10,000+ examples) but endures from 8.7% false positives and classification outlays of 18.3% CPU utilization on resource-constrained nodes. Our framework attains 96.3% detection without any requirements of training and minimal computation (6.4% of CPU). Neural network techniques [15] need offline training and scuffle with concept drift as the attack patterns evolve, while our cryptographic and threshold-based detection adapts immediately with new attack through adjustable parameters (β, θ).

7.5 Summary Table

Table 2 presents a comparative analysis of packet dropping detection schemes, where the proposed framework achieves superior detection accuracy with minimal false positives and significantly reduced overhead.

Table 2. Comparative performance of packet dropping detection (PDR) schemes

Mechanism	Detection Rate (%)	False Positive Rate (%)	Energy Overhead (%)	Computation/CPU Overhead (%)	Detection Time (s)	PDR (%)
Watchdog [1]	89.3	24.7	15.2	-	-	76.4
TWOACK [3]	93.4	7.2	-	31.4	-	88.7
EAACK [4]	94.1	12.1	27.4	-	-	89.3
CONFIDANT [6]	90.8	16.3	-	-	42–67	84.2
SVM-IDS [14]	94.2	8.7	-	18.3	-	Training required
Proposed framework	96.3	3.1	8.7	6.3	12.3	94.2

Note: Evaluated under 20% malicious nodes on a 50-node MANET.

The suggested framework illustrates higher overall performance through collaborative integration of resource awareness, cryptographic authentication, efficient ACK, and selective monitoring, gaining top-notch metrics across detection accuracy, overhead, false positive, and packet delivery ratio.

8 Conclusion and Future Work

8.1 Conclusion

This paper has introduced a comprehensive 4-phase framework for detecting and mitigating intentional packet dropping attacks in MANETs. The framework superscribes critical limitations of existing methods through four synergistic phases: resource-aware route qualification that proactively excludes honest, but constrained nodes to reduce false positive rate by 73%, password-enabled mutual authentication using CMDH-PAKE to establish cryptographically secure session keys avoiding ACK forgery, resisting authenticated digested ACKs that minimizes overhead by 85% while keeping-up detection capability, and selective on-demand indiscriminate monitoring that descends energy consumption by 42% through triggers only upon cryptographic evidence of anomalies.

Extensive NS-3 simulations demonstrate superior performance: 96.3% detection accuracy for black hole attacks and 94.7% for gray hole attacks, false positive rate of only 3.1%, packet delivery ratio maintained at 94.2% even under 40% attack intensity, end-to-end delay increase limited to 7.6%, and energy overhead constrained to 8.7% compared to baseline routing. Security analysis proves robustness against ACK forgery, impersonation, man-in-the-middle, replay, and collusion attacks. Evaluation comparison confirms that the proposed framework surpasses the modern approaches across all metrics including detection accuracy, energy consumption, communication overhead, false positives, and detection latency.

Although the framework performs well, the current implementation and evaluation are restricted to simulations and are based on lightweight pre-shared credentials for communication between nodes. For the future, scaling the framework to a large MANET, including parameter adaptation, and simulation and testbed experiments using mobility traces would be appropriate goals for further development.

8.2 Future Work

The below are some directions for future research:

• Adaptive parameter tuning: Develop machine learning techniques to dynamically adjust the thresholds and window sizes based on, traffic patterns, practical network conditions, and observed attack features.

• Cross-layer optimization: Combine the detection framework closely with physical layers and MAC layers to hold state information along with quality metrics of the link for optimized discrimination between wireless impairments and malicious drops.

• Distributed blockchain-based reputation: Explore blockchain technology to maintain tamper-proof distributed reputation ledgers that augment cryptographic detection with long-term behavioral history while preserving decentralization.

• Quality of service integration: Expand the framework to support variety of protection and detection for quality-of-service sensitive traffic classes (like A/V, emergency communications, etc.) through ranking-based monitoring and fast response methods.

• Hardware implementation: develop hardware accelerators for Chebyshev polynomial and hash-based operations to further minimize energy consumption and work on ultra-low-power IoT devices.

• Advanced attack scenarios: Explore detection mechanisms for advanced attacks including worm-hole attacks combined with packet dropping that exploit route discovery and that adjusts dropping rate based on the detection probability.

• Experimental validation: Supervise wide experiments using real devices (like smart mobile phones, embedded devices) in variety of environments to authenticate simulation results.

• Privacy-preserving detection: incorporate differential privacy techniques to prevent leakage of sensitive traffic patterns and communication metadata through control messages while maintaining detection effectiveness.

• Heterogeneous network support: Expand the framework to hybrid ad hoc networks incorporating Wi-Fi access points, cellular infrastructure, and peer-to-peer links with seamless detection across variety of network segments.

Author Contributions

Conceptualization, P.S.R. and A.A.K.M.; methodology, P.S.R.; validation, A.A.K.M.; formal analysis, P.S.R.; investigation, P.S.R. and A.A.K.M.; resources, A.A.K.M.; data curation, P.S.R.; writing—original draft preparation, P.S.R.; writing—review and editing, A.A.K.M.; visualization, P.S.R.; supervision, A.A.K.M.; project administration, A.A.K.M. All authors were actively involved in discussing the results and refining the final manuscript.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceeding 6th Annual International Conference on Mobile Computing and Networking*, Boston, United States, 2000, pp. 255–265. https://doi.org/10.1145/345910.345955

[2] K. Liu, J. Deng, P. K. Varshney, and K. Balakrishnan, "An acknowledgment-based approach for the detection of routing misbehavior in MANETs," *IEEE Trans. Mobile Comput.*, vol. 6, no. 5, pp. 536–550, 2007. https://doi.org/10.1109/TMC.2007.1036

[3] K. Balakrishnan, J. Deng, and P. K. Varshney, "TWOACK: Preventing selfishness in mobile ad hoc networks," in *Proceeding IEEE Wireless Communications and Networking Conference, 2005*, New Orleans, United States, 2005, pp. 2137–2142. https://doi.org/10.1109/WCNC.2005.1424848

[4] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS attacks in mobile ad hoc networks: A survey," in *Proceeding 2012 Second International Conference on Advanced Computing and Communication Technologies*, Rohtak, India, 2012, pp. 535–541. https://doi.org/10.1109/ACCT.2012.48

[5] N. Nasser and Y. Chen, "Enhanced intrusion detection system for discovering malicious nodes in mobile ad hoc networks," in *Proceeding 2007 IEEE International Conference on Communications*, Glasgow, UK, 2007, pp. 1154–1159. https://doi.org/10.1109/ICC.2007.196

[6] S. Buchegger and J. Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Proceeding 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Lausanne, Switzerland, 2002, pp. 226–236. https://doi.org/10.1145/513800.513828

[7] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," *arXiv preprint cs/0307012*, 2003. https://doi.org/10.48550/arXiv.cs/0307012

[8] P. Papadimitratos and Z. J. Haas, "Secure data communication in mobile ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 2, pp. 343–356, 2006. https://doi.org/10.1109/JSAC.2005.861392

[9] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceeding 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, United States, 2002, pp. 12–23. https://doi.org/10.1145/570645.570648

[10] V. Mahajan, M. Natu, and A. Sethi, "Analysis of wormhole intrusion attacks in MANETs," in *Proceeding 2008 IEEE Military Communications Conference*, San Diego, United States, 2008, pp. 1–7. https://doi.org/10.1109/MILCOM.2008.4753176

[11] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceeding IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security*, Portorož, Slovenia, 2002, pp. 107–121. https://doi.org/10.1007/978-0-387-35612-9_9

[12] S. Bansal and M. Baker, "Observation-based cooperation enforcement in ad hoc networks," *arXiv preprint cs/0307012*, 2003. https://doi.org/10.48550/arXiv.cs/0307012

[13] S. Buchegger and J. Y. L. Boudec, "A robust reputation system for P2P and mobile ad-hoc networks," in *Proceeding Second Workshop Economics of P2P Systems*, Philadelphia, United States, 2004.

[14] N. A. Hikal, M. Y. Shams, H. Salem, and M. M. Eid, "Detection of black-hole attacks in MANET using adaboost support vector machine," *J. Intell. Fuzzy Syst.*, vol. 41, no. 1, pp. 669–682, 2021. https://doi.org/10.3233/JIFS-202471

[15] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C. Y. Tseng, and T. Bowen, "A general cooperative intrusion detection architecture for MANETs," in *Proceeding Third IEEE International Workshop on Information Assurance (IWIA'05)*, College Park, United States, 2005, pp. 57–70. https://doi.org/10.1109/IWIA.2005.1

[16] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *IEEE Wireless Commun.*, vol. 11, no. 1, pp. 48–60, 2004. https://doi.org/10.1109/MWC.2004.1269717

[17] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceeding 6th Annual International Conference on Mobile Computing and Networking*, Boston, United States, 2000, pp. 275–283. https://doi.org/10.1145/345910.345958

[18] V. Balakrishnan, V. Varadharajan, and U. K. Tupakula, "Fellowship: Defense against flooding and packet drop attacks in MANET," in *Proceeding 2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006*, Vancouver, Canada, 2006, pp. 1–4. https://doi.org/10.1109/NOMS.2006.1687659

[19] L. Venkatraman and D. P. Agrawal, "A novel authentication scheme for ad hoc networks," in *Proceeding 2000 IEEE Wireless Communications and Networking Conference*, Chicago, United States, 2000, pp. 1268–1273. https://doi.org/10.1109/WCNC.2000.904814

[20] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Netw.*, vol. 13, no. 6, pp. 24–30, 1999. https://doi.org/10.1109/65.806983

[21] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Proceeding International Conference on the Theory and Application of Cryptology and Information Security*, Amsterdam, Netherlands, 2002, pp. 548–566. https://doi.org/10.1007/3-540-36178-2_34

[22] X. Ding, F. Wei, C. Ma, and S. Chen, "An efficient password authenticated key exchange protocol with bilinear parings," in *Proceeding International Conference on Information Security and Assurance*, Seoul, Korea, 2009, pp. 50–56. https://doi.org/10.1007/978-3-642-02633-1_7

[23] L. Kocarev and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*. Springer, 2011.

[24] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, no. 12, pp. 4052–4057, 2010. https://doi.org/10.1016/j.cnsns.2010.02.014