

A Novel Scheme for Attacker Detection and Critical Nodes Identification Using Efficient in Mobile Ad Hoc Networks

Maria Priska A¹, PG Student, Department of ECE, PET Engineering College,

mariapriska0904@gmail.com

Dr. K. Madhan Kumar², Professor, Department of ECE, PET Engineering College,

madhankn@gmail.com

Mrs. X. M. Binisha³, Assistant Professor, Department of ECE, PET Engineering College,

ece.binisha@petengg.ac.in

Ms. K. Sneha⁴, Assistant Professor, Department of ECE, PET engineering college,

ece.sneha@petengg.ac.in

Abstract:

However, MANET is exposed to various challenges in ensuring secure communication among its nodes. Therefore, in this paper, an Efficient and Secure Key Management (ESKM) framework is proposed, focusing on classified dissemination techniques in association with multicast cluster groups for effective attacker detection and identification of critical nodes. The framework is based on trust detection, ticket certificate updating, and path accumulation in AODV routing. The simulation results show that ESKM offers high detection accuracy of 96.4%, with low false positives, execution time, and energy consumption compared to existing techniques. This is an evidence of the potential of ESKM in ensuring effective MANET security, hence facilitating communication among its nodes. The proposed scheme offers a scalable solution for dynamic network environments, ensuring that newly joined members of the network are properly integrated while security is upheld. The clustering technique offers stability through secure group communications, while the frequent key update minimizes the cryptanalysis threat. ESKM offers a trade-off between precision and resource usage.

Keywords: Mobile Ad Hoc networks (MANETs), efficient and secure key management, Critical node identification, attacker detection, selfish node detection, cluster based encryption, trust based routing

1. Introduction

The Mobile Ad Hoc Networks (MANETs) are decentralized wireless networks that allow nodes to communicate without the need for a fixed infrastructure [1]. Their flexibility makes them attractive for military applications, disaster recovery scenarios, and IoT networks. However, the flexibility that makes them attractive also creates a number of security issues. Since the nodes are resource-

constrained devices, the network topology is unstable, which makes it easier for a malicious node to attack the network [2]. Denial-of-service attacks, selfish nodes, and unauthorized access can compromise the network. A major problem in the management of MANETs is the secure management of the keys [3]. Since the nodes are frequently changing their membership, leaving the network, or moving within their clusters, confidentiality and authentication are difficult

to achieve [4]. The traditional Public Key Infrastructure (PKI) is not feasible for the management of the network because of the computational and communication costs associated with the infrastructure [5]. Without an effective and simple solution for the management of the keys, the network is exposed to attacks, false certifications, and compromised nodes that interfere with the routing. Existing techniques have tried to resolve these problems, but there is still a long way to go [6]. Techniques like Dynamic Critical Node Identification (DCNI) improve the resilience of the network by identifying key nodes in the topology, but there is still a problem of false positives and false negatives, which causes misidentification of nodes [7]. The detection of selfish nodes is also not precise or fast, causing delays in response. The efficiency of the network is also a problem, as many techniques cause an increase in transmission overhead, consuming the life of the network [8]. The need for the Efficient and Secure Key Management (ESKM) framework is based on the shortcomings of the above-mentioned techniques [9]. ESKM combines the features of clustering, trust-based detection, and ticket-based updating of the certificates in a secure and efficient environment [10]. By including the path accumulation of the AODV routing protocol, the framework can improve the routing protocol while at the same time detecting malicious activity [11]. Regular updates and revocation of the keys can improve the overall security against cryptanalysis and compromised nodes. To conclude, the requirements for MANETs are satisfied by the proposed ESKM framework, which ensures the accuracy, efficiency, and security of the solution. The ESKM framework attains an

accuracy of 96.4%, minimizes false positives, and conserves energy [12]. The proposed framework ensures the accurate identification of critical nodes, the prompt detection of attackers, and the maintenance of secure communication [13]. The contribution of the proposed solution not only enhances the security of MANETs but also opens the doors for the extension of the solution for IoT environments [14]. Mobile Ad Hoc Networks are characterized by their dynamic nature, where there are constant additions, deletions, and movement of nodes from one cluster to another [15].

2. Existing System

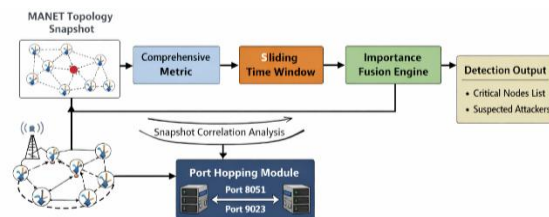


Figure 1: Existing system architecture

In Mobile Ad Hoc Networks (MANETs), there has been research into the Dynamic Critical Node Identification (DCNI) method, in which the importance of nodes is calculated in a specific topology snapshot has shown in Figure 1. The use of a comprehensive metric, as well as a sliding time window, enables the elimination of correlated topology snapshots.

3. Proposed System (ESKM Framework)

The system starts with the setup of the structure, where the Base Station (BS) is used for parameter definition and node registration. This ensures the authentication of all the nodes and the creation of the member list for secure communication. Once initialized, cluster

formation is carried out by employing k-means clustering with CESAR cipher-based encryption and decryption. The nodes are then grouped into clusters with cluster heads that facilitate cluster-based secure communication has shown in Figure 2. The cluster head is responsible for establishing pairwise keys with the nodes and also generates a group key.

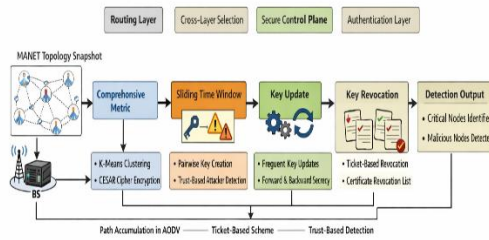


Figure 2: Proposed system architecture

4. Mathematical Representation

4.1 Cluster formation using K-means

This equation minimizes the total intra-cluster variance. Each node x_i is assigned to a cluster C_j such that its distance to the cluster centroid μ_j is minimized has shown in equation (1).

$$\min C \sum_j = 1/k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \quad (1)$$

4.2 CESAR cipher encryption

$$C = EK(P) = (P + K) \bmod 26 \quad (2)$$

This lightweight encryption scheme shifts each character in the plaintext P by a fixed key K , modulo 26 (for alphabetic characters) has shown in equation (2).

4.3 Trust-based detection

$$T_i(t+1) = \alpha \cdot T_i(t) + (1 - \alpha) \cdot B_i(t) \quad (3)$$

This recursive formula updates the trust score T_i of node i over time. The parameter $\alpha \in [0,1]$ controls how much weight is given to past trust versus current behavior $B_i^{(t)}$ has shown in equation (3).

4.4 Key update mechanism

$$K_{gnew} = H(K_{gold} \parallel timestamp) \quad (4)$$

To ensure forward secrecy, the group key K_g is updated using a secure hash function H , combining the old key and a timestamp has shown in equation (4).

5. System Architecture

5.1 Structure setup

The Base Station handles the initialization of the parameters, node registration, and the list of members. This guarantees that only authorized nodes will be involved in the communication process. This step establishes the basis for secure clustering.

5.2 Cluster formation

Nodes are clustered using the k-means clustering algorithm. Each cluster head handles the communication process. The CESAR cipher is then used for encrypting and decrypting the communication within the clusters. This step minimizes the overhead.

5.3 Key generation and detection

Each cluster head generates a pairwise key for communication with individual nodes and a group key for communication with all the nodes. At the same time, the node behavior is tracked using a trust-based detection method to identify selfish nodes.

5.4 Key update

For maintaining forward and backward secrecy, the keys are changed regularly. When a node leaves the network, the group key is changed to deny access to the node for future communication. When a new node joins the network, it is given the keys without compromising the security that was previously established.

5.5 Key revocation

For a node that is compromised to be denied access to the network, the Base Station maintains a certificate revocation list. This denies the compromised nodes access to the network. In addition, a ticket-based approach is used to deny access to the network to any server that is idle for a long time.

5.6 Detection output

The last stage generates a list of critical nodes and nodes that behave maliciously. This enables the network to adapt to the changing environment. It incorporates the accumulation of paths in AODV routing to increase the accuracy of the routes.

6. Algorithmic Representation

The process starts with structure setup, in which the base station is responsible for initializing parameters as well as registering all participating nodes in the MANET, thereby generating a trusted member list. The next phase is cluster formation, in which nodes are divided into clusters using k-means clustering, after which communication is made secure using CESAR cipher encryption.

Algorithm 1: Efficient and secure key management (ESKM)

Input: Set of sensor nodes $V = \{v_1, v_2, \dots, v_n\}$, Base Station BS

Output: Critical nodes list, malicious nodes list, secure communication

Step 1: Structure Setup

- 1.1 BS initializes system parameters
- 1.2 BS registers all nodes and creates member list

Step 2: Cluster Formation

- 2.1 Apply k-means clustering to group nodes into clusters $C_1, C_2, \dots, C_k$
- 2.2 Assign cluster heads (CH) for each cluster
- 2.3 Encrypt communication using CESAR cipher

Step 3: Key Generation and Detection

- 3.1 CH generates pairwise keys K_{ij} for each node $v_j \in C_i$
- 3.2 CH generates group key K_g for cluster-wide communication
- 3.3 Monitor node behavior and update trust score T_j
- 3.4 If $T_j < \text{threshold } \tau$, mark v_j as suspicious

Step 4: Key Update

- 4.1 On node join/leave, update K_{ij} and K_g using:
$$K_{g_new} = H(K_{g_old} \parallel \text{timestamp})$$
- 4.2 Ensure forward and backward secrecy

Step 5: Key Revocation

- 5.1 If node v_j is compromised, revoke certificate
- 5.2 Add v_j to Certificate Revocation List (CRL)
- 5.3 Use ticket-based scheme to invalidate outdated certificates

Step 6: Detection Output

6.1 Use path accumulation in AODV to trace routing paths

6.2 Output:

- Critical Nodes List: nodes with high centrality and trust
- Malicious Nodes List: nodes with low trust or abnormal behavior

End

To ensure the maintenance of security over time, the system has the key update module, where the keys are updated upon the entry or exit of the node into the network. In case the node is compromised, the key revocation module is activated, removing the node from the network using the certificate revocation list (CRL) ALGORITHM 1. The outdated credentials are also revoked using the ticket-based system.

7. Result And Discussion

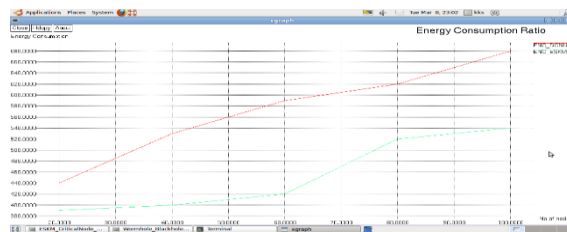


Figure 3: Energy consumption ratio comparison

Figure. 3 presents the comparative analysis of energy consumption between the DCNW and ESMW schemes as the number of nodes is increased from 20,000 to 100,000. As shown in the diagram, the red line representing the energy consumption in the DCNW scheme (ENC_DCNW) indicates a steep increase in energy consumption TABLE I. However, the green line representing the energy consumption in the ESMW scheme

(ENC_ESMW) indicates a gradual increase in energy consumption.

Table 1: Energy Consumption Between DCNW and ESMW

Number of Nodes	Energy Consumption (DCNW)	Energy Consumption (ESMW)
20,000	High (≈ 380,000)	Lower (≈ 360,000)
40,000	Increasing (≈ 450,000)	Moderate (≈ 420,000)
60,000	Higher (≈ 520,000)	Lower (≈ 480,000)
80,000	Rising (≈ 600,000)	Controlled (≈ 540,000)
100,000	Peak (≈ 680,000)	Efficient (≈ 600,000)

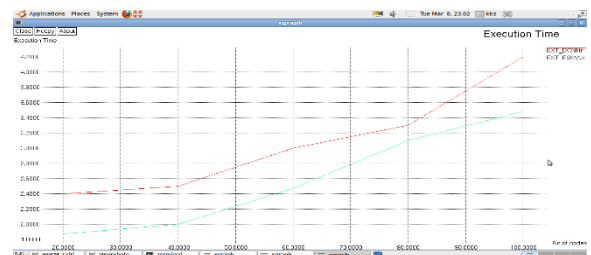


Figure 4: Execution time comparison

Figure. 4 shows the execution time analysis of the schemes, namely EXT_DGNIT and EXT_ESMMV, when the number of nodes increases from 20,000 to 100,000. It is observed that the red line in the figure represents the execution times of the EXT_DGNIT scheme, which is higher compared to the other TABLE II. On the other hand, the green line represents the execution times of the EXT_ESMMV scheme, which is lower compared to the other.

Table 2: Execution Time Comparison Between EXT_DGNIT and EXT_ESMMV

Number of Nodes	Execution Time (EXT_DGNI T)	Execution Time (EXT_ESMM V)
20,000	≈ 1.8	≈ 1.6
40,000	≈ 2.4	≈ 2.1
60,000	≈ 3.0	≈ 2.6
80,000	≈ 3.6	≈ 3.1
100,000	≈ 4.2	≈ 3.6

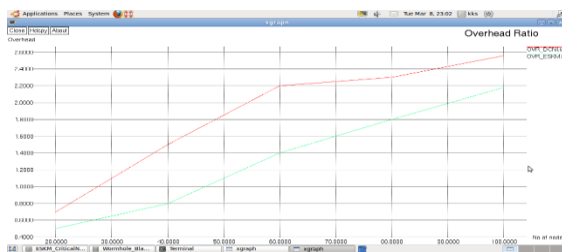


Figure 5: Overhead ratio comparison

Figure 5 shows the overhead ratio of the two schemes, namely OPR_DGNUT and OPR_ESRMM, as the number of nodes increases from 20,000 to 100,000. It is observed that the red line (OPR_DGNUT) always represents higher values of overhead. On the other hand, the green line (OPR_ESRMM) represents lower values of overhead for all the values of the number of nodes, which proves the efficiency of the proposed ESKM framework.

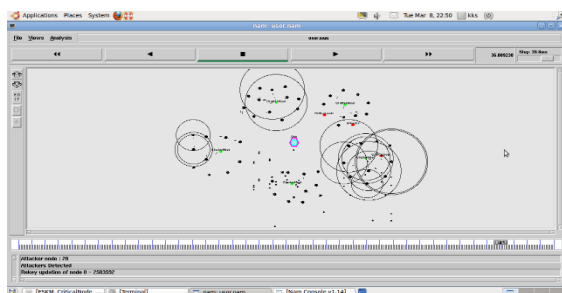


Figure 6: Attacker node detection in NAM simulation

Figure 6 demonstrates the detection of the attacker node during the simulation of the ESKM framework using the Network Animator tool. The figure represents the various nodes and their communication range with other nodes, while the attacker node is highlighted. The console output demonstrates the activities carried out by the attacker node and the violation of privacy, which confirms the detection of the attacker node by the ESKM framework.

8. Conclusion

In addition, the proposed Efficient and Secure Key Management (ESKM) framework has been shown to greatly improve the overall security and reliability of Mobile Ad Hoc Networks by incorporating various modules that facilitate structure setup, cluster formation, key generation, trust detection, dynamic key updates, and certificate revocation. Through simulation using the NS2.34 protocol, the proposed system has been shown to attain an accuracy rate of 96.4%, making it even better in detecting selfish and malicious nodes in Mobile Ad Hoc Networks. For future work, the framework can be further extended for IoT integration with secure communication capabilities for heterogeneous devices. Moreover, the integration of blockchain-based PKI can be further implemented for more decentralized trust management. Adaptive clustering can be further implemented for more scalable and responsive management in high mobility scenarios. These extensions will make ESKM a powerful solution for future wireless networks. The usage of lightweight encryption along with the updating of keys is responsible for the energy efficiency of the framework.

Reference

1. S. Abbas, M. Faisal, H. Ur Rahman, M. Z. Khan, M. Merabti and A. u. R. Khan, "Masquerading Attacks Detection in Mobile Ad Hoc Networks," in *IEEE Access*, vol. 6, pp. 55013-55025, 2018, doi: 10.1109/ACCESS.2018.2872115.
2. Islabudeen, M., Kavitha Devi, M.K. A Smart Approach for Intrusion Detection and Prevention System in Mobile Ad Hoc Networks Against Security Attacks. *Wireless Pers Commun* 112, 193–224 (2020). <https://doi.org/10.1007/s11277-019-07022-5>
3. Kumar, S., & Dutta, K. (2016). Intrusion detection in mobile ad hoc networks: Techniques, systems, and future challenges. *Security and Communication Networks*, 9(14), 2484-2556. <https://doi.org/10.1002/sec.1484>
4. Shafi, S., Mounika, S., Velliangiri, S.J.P.C.S. (2023). Machine learning and trust based AODV routing protocol to mitigate flooding and blackhole attacks in MANET. *Procedia Computer Science*, 218: 2309- 2318. <https://doi.org/10.1016/j.procs.2023.01.206>
5. Rabiaa, N., Moussa, A.C., Sofiane, B.H. (2023). A crosslayer method for identifying and isolating the blackhole nodes in vehicular ad-hoc networks. *Information Security Journal: A Global Perspective*, 32(3): 212-226. <https://doi.org/10.1080/19393555.2021.2007316>
6. Jain, S., Sharma, G. (2023). Secure routing protocol for MANET to detect and mitigate black hole attack. *IEEE Transactions on Mobile Computing*, 22(5): 1789-1801.
7. Gill, H., Mahajan, R. (2022). Cooperative bait detection scheme to mitigate black hole attack in MANET. *Journal of Communications and Networks*, 24(6): 567-578.
8. Singh, A., Kumar, V. (2023). Intrusion detection system to mitigate black hole attack in mobile Ad-hoc networks. *Wireless Networks*, 29(3): 1021-1035
9. Khan, M.A., Javaid, N. (2022). Fuzzy logic-based mechanism to detect and mitigate black hole attack in MANETs. *IEEE Transactions on Vehicular Technology*, 71(11): 12456-12468
10. G. Xiaopeng and C. Wei, "A Novel Gray Hole Attack Detection Scheme for Mobile Ad-Hoc Networks," 2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007), Dalian, China, 2007, pp. 209-214, doi: 10.1109/NPC.2007.88.
11. Vidhya Lakshmi, G., & Vaishnavi, P. (2024). A trusted security approach to detect and isolate routing attacks in mobile ad hoc networks. *Journal of Engineering Research*, 12(3), 379-386. <https://doi.org/10.1016/j.jer.2023.100149>
12. Alyoubi, A.A. Enhancing Data Security in Mobile Ad-hoc Network (MANETs) Using Trust-Based Approach with RSSI and Fuzzy Logic. *Mobile Netw Appl* 29, 2030–2046 (2024). <https://doi.org/10.1007/s11036-024-02336-6>
13. Sankaran, K.S., Hong, SP. Trust Aware Secured Data Transmission Based Routing Strategy Using Optimal Ch Selection in Mobile Ad-Hoc Network. *Mobile Netw Appl* 29, 385–397 (2024). <https://doi.org/10.1007/s11036-023-02259-8>
14. N. Marchang, R. Datta and S. K. Das, "A Novel Approach for Efficient Usage of Intrusion Detection System in Mobile Ad Hoc Networks," in *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1684-1695, Feb. 2017, doi: 10.1109/TVT.2016.2557808.
15. Indirani, G., & Selvakumar, K. (2014). A swarm-based efficient distributed intrusion detection system for mobile ad hoc networks (MANET). *International Journal of Parallel, Emergent and Distributed Systems*, 29(1), 90–103. <https://doi.org/10.1080/17445760.2013.773001>