Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.030444
Article

Tech Science Press

Hybrid Authentication Using Node Trustworthy to Detect Vulnerable Nodes

S. M. Udhaya Sankar[1,*], S. Thanga Revathi[2] and R. Thiagarajan[3]

[1]Department of Information Technology, Velammal Insitutue of Technology, Chennai, 601204, India
[2]Department of Networking and Communication, SRM Institute of Science and Technology, Kattankulathur, 603203, India
[3]Department of Information Technology, Prathyusha Engineering College, Thiruvallur, 602025, India
*Corresponding Author: S. M. Udhaya Sankar. Email: udhaya3@gmail.com
Received: 24 March 2022; Accepted: 26 April 2022

Abstract: An ad-hoc sensor network (ASN) is a group of sensing nodes that transmit data over a wireless link to a target node, direct or indirect, through a series of nodes. ASN becomes a high-risk group for several security exploits due to the sensor node's limited resources. Internal threats are more challenging to protect against than external attacks. The nodes are grouped, and calculate each node's trust level. The trust level is the result of combining internal and external trust degrees. Cluster heads (CH) are chosen based on the anticipated trust levels. The communications are then digitally signed by the source, encoded using a key pair given by a trustworthy CH, decoded by the recipient, and supervised by verifications. It authenticates the technique by identifying the presence of both the transmitter and the recipient. Our approach looks for a trustworthy neighboring node that meets the trust threshold condition to authenticate the key produced. The companion node reaffirms the node's reliability by getting the public-key certification. The seeking sensor node and the certification issuer node must have a close and trusting relationship. The results of the proposed hybrid authentication using a node trustworthy (HANT) system are modeled and tested, and the suggested approach outperforms conventional trust-based approaches in throughput, latency, lifetime, and vulnerability methods.

Keywords: *Ad hoc* sensor network; wireless security; clustering; cryptography; key management

1 Introduction

A wireless sensor network (WSN) is an *ad hoc* network that detects environmental and physical elements such as heat, audio, movement, vibrations, and pressure. Therefore, we cannot exaggerate the importance of safety in the route-finding process. As a result, WSNs' power, throughput, and storage capacity are limited, and security methods' deployment is severely restricted. The security measures are access control, authenticity, node validation, and honesty. Security mechanisms designed for all other systems do not operate on WSNs [1]. As a result, it requires new safety measures to fight against threats. Distributing the dealing with large amounts of sensor data improves sensing accuracy—each sensor node with in-network functions independently, with no centralized administration location. The node's decisions rely on its purpose, the information provided, and the expertise of its processing, transmission, and energy supplies.

Future's new networked sensors can function with greater accuracy, resilience, and intelligence than current isolated devices. Until this vision may become a fact, it must address many hurdles. Though WSN has a lot of potential for a broad variety of applications, the difficulty of protecting them has been a bottleneck to their wider acceptance and implementation. The research area of protecting WSN is still in its early stages. While WSNs are vulnerable to the same security issues as traditional networks, they also experience numerous threats. It depends on the physical features of sensor devices, such as short transmission, channel capacity, computational capability, storage, and limited battery life, implementation atmosphere. In WSN, where connectivity sensor nodes have restricted network bandwidth, Computational power, storage, and battery life, conventional safety procedures for maintaining secrecy, authenticity, and availability are ineffective.

Fig. 1 depicts a Wireless ASN in which every mobile host is the liberty to move in any position, resulting in frequent connectivity fluctuations. It comprises a self-contained infrastructure of movable nodes that can join and detach from any other. Every mobile host performs data collection and routers to transfer datagrams, and the ASN nodes change dynamically. Because of its periodically distributed architecture and flexibility, it may use ASN in various fields.

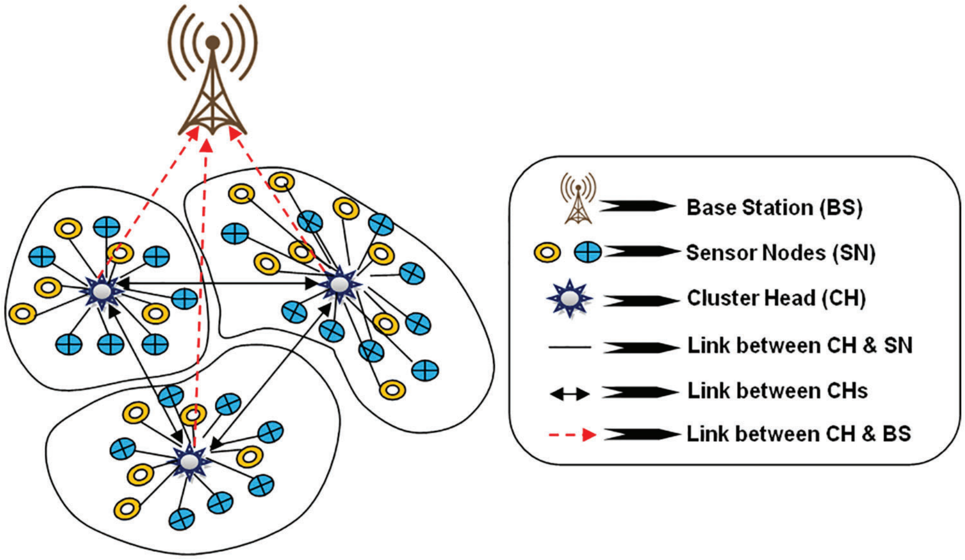

Figure 1: WASN architecture

Due to poor interconnection, resource limits, and the movable nodes' minimal physical defense, security is an essential issue in WASN. WSNs are thus more prone to damage than infrastructure-based networking. Because of the:

1. It shares a wireless channel.
2. No clear line of protection.
3. Self-organizing and dynamic network.
4. Mostly broadcast communications.
5. Messages transport hop-by-hop.
6. Sensor nodes are restricted when it comes to processing power and battery capacity.

Constructing cryptographically secure ASN is notoriously problematic. By evaluating the integrity of its neighbors beforehand or in real-time to make any forwarding decisions, reputation can recreate a crucial role in enhancing the security of *ad hoc* networks. Due to resources (e.g., power, capacity) limits, all nodes may be unable to participate in navigation, and, as a result, it is not forwarding the datagrams.

A node's trustworthiness is essential for ensuring node availability and providing secure node-to-node interactions between them. Cryptographic techniques cannot detect or prevent such unpredictable activity that poses a security risk to the connection. The concept of trust evaluation complements rather than replaces cryptography. To accomplish holistic security in ASN, cryptographic and a reputation determination framework can function together in a hybrid manner [2]. The current study has established and implemented a substantiation trust architecture in a cluster-based ASN to assure trustworthy, safe, and timely throughput in the network. The ASN nodes are self-organized into one-hop clusters for trust computation and dissemination, and we build a forecast trust evaluation model. This approach works effectively for identifying vulnerable nodes and boosting system performance.

The remaining paper follows: Section 2 provides a synopsis of relevant MANET research activities in trustworthiness and secure routing. Then, we describe our proposed hybrid trust framework in Section 3; next, the simulation analysis and outcomes of our trust mechanism, and other previous protocols, are shown in Section 4; at last, we examine the conclusions in Section 5.

2 Related Works

The researchers presented trust-based secure wireless protocols for various networks, including social media platforms, ASN, peer-to-peer networks, and WSN. Ishmanov et al. [3] describe the trust management mechanism in WSN clearly. They highlighted the impact of trust management in WSNs, contrasted several types of trust evaluation, and identified several possible research questions, including surveillance and monitoring, trust assessment, trust replication, attack tolerance, trust administration, system performance comparisons. The trust mechanism is emerging, and it needs to be improved in various ways because WSNs is still a growing field it uses trust mechanism. As a result, the trust evaluation procedure plays a significant function in ensuring safe data transmission. To efficiently cope with selfishness or fraudulent nodes, in paper [4] present a highly scalable cluster-based multilevel reputation framework for WSN. Several factors determine the level of trust.

In [5], the authors suggest a reputation-based paradigm for data truthfulness. The recommended reputation process utilizes data gathered from each node to identify erroneous data and unfriendly nodes using a watchdog function. In [6] presented a parametric and localized trust managing model for WSN security, namely safe routing, in which each node keeps a strongly abstracted variable to assess its neighbors. In [7] propose a trust-aware secure routing architecture. Researchers have created a model for evaluating a routing protocol trust relationship. In paper [8] suggested A list of standard practices for making a good reputation management platform for WSNs. In [9] recommend, a trust evaluation paradigm based on events. The trust is estimated based on the event occurring and the confidence level. Most trust models concentrate on a single sort of trust evaluation application. Therefore, it is critical to establish a platform in which the reputation management solution is evaluated at several protocol layers and can be used to recognize various types of threats.

For diverse kinds of keys (pairs keys, cluster keys, individual keys) [10] and distinct types of social topology, there is a range of solutions addressing key exchange protocol in WSNs (hierarchical, flat). There were proposals for both symmetrical and asymmetric key-based systems. We concentrate on the most effective procedures, the symmetric-key-based individuals, for performance reasons. We also confine our analysis to WSAN key management methods that address safety at the roup stage, excluding strategies like SPINS [11] and BROSK [12] that concentrate on pairwise key creation. The Localized

Encryption and Authentication Protocol (LEAP) [13] are among the most comprehensive symmetrical key distribution techniques for WSANs. This hierarchical scheme creates a pairwise key across CH, a batch key inside the identical group, and a networking key. [14–16] discuss various symmetric-key-based cluster key managing strategies for the tree-and star-based systems. The group key is built-in [14] using the assessment of a bivariate polynomial and Lagrangian interpolation. A generating matrix uses to create the group key [15,16].

There exist merely a few symmetric-key-based ideas for key exchange in dispersed ad-hoc systems with arbitrarily deployed nodes. It is more challenging because the locations of a node within the network are still not known at the time. The keys in [17] create using a robot-assisted networking bootstrapping procedure. Another solution has recently been presented in [18], which uses a Network Multicast Manager (NMM), which is cloud-based and a specialized asymmetrical transmission of the actual key via NMM and a node with sensing capability to create multicast keys across a dynamic cluster of nodes. However, the focus of all of these systems is on generating a group key. Occasionally, it supplements a pair of keys among nodes and an autonomous key with the CH/BS. The authenticity of particular sensor nodes verified by all other group members is still to be solved using cryptography. In [19], authors examine trustworthiness and reputation-based methods, which generally involve a lot of computing effort and establish primarily at the gateways level. Despite the work required, the functionality is worthwhile to investigate, especially for cooperative WSANs [20]. For starters, it enables the node to check the honesty of its neighbors so it can use that information captured from others for its objectives (e.g., calibration). Second, it facilitates the detection of fraudulent nodes within the network and the prevention of all varieties of a sinkhole, wormholes, sleep, Sybil, selective forwarding, and denial of services.

Finally, many statistics [21] and Machine learning approaches [22] represents in the literature for identifying suspicious nodes. Unfortunately, despite being the most effective resource for avoiding these types of attacks, a system based on a cryptographic system is not yet accessible in the research. The TESLA method [1] and its versions [23–25] are usually the most popular method for clustering with node authentication utilizing symmetric key cryptography. However, these systems rely on a hashing chain and disclose each chain's values in predetermined time intervals. As a result, there is a slot latency in verifying the message's legitimacy. Then there is the [26] method, which uses a hashing chain but combines it with authentication and encryption, leading to a system with no authentication latency. Finally, Bertoni et al. developed a technique implemented in an edge-based framework [27]. We present a hybrid authentication protocol for all sensing nodes in the system in this study, which relies on an expandable output feature based on reputation mechanisms.

3 System Model

3.1 Overview of HANT Model

This research presents a hybrid trust method based on node authentication for cluster-based ASNs. First, the sensor nodes are grouped in this way [28]. Then, We do each node's trustworthiness level assessment directly and indirectly. Prior interactions with neighbors determine the direct trust level and the indirect trust level by recommended trust level from its most comparable nearest neighbors. Next, the cluster heads (CH) are chosen based on the projected trust level. After then, a group of verification keeps an eye on each CH. The communications are then digitally signed by the transmitter, encrypted with a key pair delivered by a trusted party, and decoded by the recipient. Finally, it authenticates the strategy by verifying the identity of both the transmitter and the recipient.

The suggested trust-based authentication mechanism for clustered ASNs depict in Fig. 2.

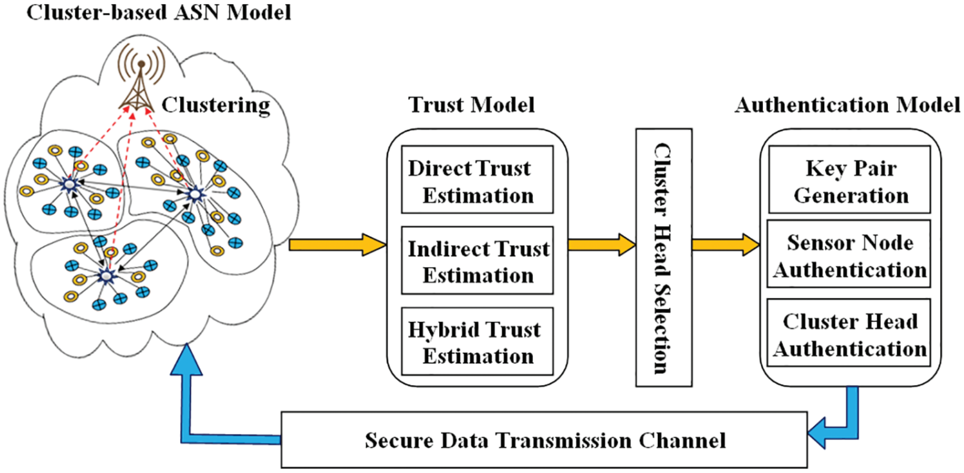

Figure 2: HANT architecture

3.2 Clustering of Sensor Nodes for CH Selection

We suppose the network has several Certificate authorities (CAs), each of which can authenticate all nodes inside its jurisdiction. CA is a trustworthy third party in charge of Node IDs, encryption keys, and rights. The nodes first divide into several clusters with the same frequency classes, each with 1 CH in a dynamic situation. Each set consists of a CH and one or more nodes [29–31]. Nodes in much the same cluster are connected, while nodes in other groups can communicate with one another via their CHs. Each node can function as a CH, gateway, or client. A gateway is a node that connects two or more groups, and it is the responsibility of each CH to keep track of its participants and gateways. Algorithm 1 explains the CH election procedure. Tab. 1 lists the different notations and their meanings.

Table 1: Notations and meaning

Notations	Meaning
ɳi	Each Node in the ASN, i = l, 2, 3……….
₵ɳi	Connecting Neighbor of ɳi
ⱡi	Location of ɳi
ɨ^{đ}i	Identity of ɳi
₵ⱡi	Connecting Neighbor list of ɳi
$D_{i\rightarrow j}$	Distance between ɳi and ɳj
^{И}ɳi	Number of neighbors of ɳi
℟†	Dynamic transmission range
Ọ	Direction of Node
V̏	Node Velocity
Ťɖ	Degree of Trust
₩i	Weigh of Node
Ť^{ᵖ}	Direct Trust Degree
Ť$_{i\rightarrow j}$	Degree of Trust between ɳi and ɳj

(Continued)

Table 1 (continued)

Notations	Meaning
$\text{Ť}^{\text{ɨ}}$	Indirect Trust Degree
$\forall^{\text{ꞩ}}$	For all Successful Transmission
ɆꞘ	For all Failure Transmission
$\sum_{i=1}^{n}\text{Ⱶ}$	Honor Point for every successful transmission Behavior, Where n = 0, 1, 2…..
$\sum_{i=0}^{-n}\text{Я}$	Retribution Point for every failed transaction behavior, where n = 0, -1, -2, -3…..
wa, wb, wc, wd	Weight constants (0 to 1)
Ťm	Trust minimum
Ṝ	Reputation of node
Ꞔǎ	Certificate authority
$\dot{\check{\text{S}}}^{\text{ᵖ}}_{i\rightarrow k}$	Direct similarity between node i and k
ṱ and Δṱ	Time and Delay Time

Algorithm 1

```
Step 1: Each node ɳi declare itself a CH and broadcasts the RB[ŧi, ɨᵭi] radio beacon.
Step 2: After getting RB[ŧi, ɨᵭi] from each ɳi, each node ɳi builds ₵ŧi.
Step 3: Finally, ɳi calculates $\text{Ɗ}_{i\rightarrow j}$.
Step 4: ɳi generates a weighted total in step four.
Step 5: Find the weight using Eq. (1)
```

$$\text{₩i} = \sum_{i=0}^{n} w_a \text{Ͷ}\text{ɳ}_i + w_b \text{Ɽ†} + w_c \text{♀} + w_d \text{V̏} - \text{Ťɗ} \tag{1}$$

The node calculates the components required for Eq. (1). It will use weighted constants from the 0 to 1 range. Because the weighting factor is determined using these criteria, the CH chosen will be the most trustworthy and effective. Then, node with minimal weight is chosen as CH.

3.3 Trust Model

CH are chosen based on their degree of trustworthiness. Direct trust is a trust connection formed via direct neighbor interaction. Indirect trust is a relationship created by a neighboring node or a network of trusted nodes. The computation of direct trust level between node-i and it neighbor node-j as follows:

$$\check{\text{T}}^{\text{ᵖ}}_{i\rightarrow j} = \begin{cases} \sum_{i=1}^{n} \text{Ⱶ}_{i\rightarrow j} + \sum_{i=1}^{n} \check{\text{T}}_{i\rightarrow j} & (\forall^{\text{ꞩ}} > 0) \\ \sum_{i=1}^{-n} \text{Я}_{i\rightarrow j} - \sum_{i=1}^{n} \check{\text{T}}_{i\rightarrow j} & (\text{ɆꞘ} > 0) \end{cases} \tag{2}$$

where Ťi → j trust degree from node-j to node-i (i.e., the value calculated during the previous CH selection process). Also, Ⱶi → j is the honor points given to node 'i' by node j for every $\forall^{\text{ꞩ}}$, and Яi → j is the retribution points given to node 'i' by node j for all ɆꞘ.

We calculate the indirect trust level between nodes i and j as follows:

$$\check{T}^{\text{ɨ}}_{i\rightarrow j}=\frac{\sum_{k\in \text{ɳ}i}^{n}\check{T}^{\text{ᴅ}}_{k\rightarrow j}\times \dot{\check{S}}^{\text{ᴅ}}_{i\rightarrow k}}{\sum_{k\in \text{ɳ}i}^{n}S^{?}_{i\rightarrow k}} \tag{3}$$

where,

$$\dot{\check{S}}^{\text{ᴅ}}_{i\rightarrow k}=\frac{\sum_{i=1}^{n}\check{T}\text{ɖ}_{i}*\check{T}\text{ɖ}_{k}}{\sqrt{\sum_{i=1}^{n}\check{T}\text{ɖ}_{i}^{2}}*\sqrt{\sum_{k=1}^{n}\check{T}\text{ɖ}_{k}^{2}}} \tag{4}$$

The estimation of reputation level is the combine value of direct and indirect trust level.

$$R=w_{a}*\check{T}^{\text{ᴅ}}_{i\rightarrow j}+w_{b}*\check{T}^{\text{ɨ}}_{i\rightarrow j} \tag{5}$$

Algorithm 2

```
Step 1: Node ‘i’ gather the knowledge of the local topologies.
Step 2: Node ‘i’ uses ₵ɳi to find $\check{T}^{\text{ᴅ}}_{i\rightarrow j}$ rely on the neighbor’s list and previous occurrences by Eq. (2).
Step 3: If the i and j don’t communicate, move on to the next step.
Step 4: Ťi ⟶ j = $\check{T}^{\text{ᴅ}}_{i\rightarrow j}$
Step 5: Save $\check{T}^{\text{ᴅ}}_{i\rightarrow j}$ in the local data table.
Step 6: End If.
Step 7: If the variables i and j connect, then
Step 8: It should update $\check{T}^{\text{ᴅ}}_{i\rightarrow j}$
Step 9: $\check{T}^{\text{ɨ}}_{i\rightarrow j}$ is determined by i to j using equivalent $\check{T}^{\text{ɨ}}_{i\rightarrow j}$ and $\check{T}^{\text{ɨ}}_{i\rightarrow j}$ values using Eq. (3).
Step 10: Using the formula (4) calculate R
Step 11: End If
```

3.4 Authentication Model

Initially, we presume that authentic nodes want to join the ASNs by giving public/private, shared key, and certificate. The keys can be physically input or transferred using secure methods. Digital certificates (DC) use to protect data sent by a node. Every control message ends with a DC from the originator. The DC makes the signatory's values and the content for authentication. The sender's secret key and the destination verify the communication sign with the signer's public key. The nodes requesting verification give the validating node its identification and certificates during the authentication phase [32–34]. After verifying the certification with the CA's public key, the certifying node will keep challenging the originating node by encoding a nonce with the originating node's shared key to see if it has the appropriate secret keys. After the interaction, two nodes transfer the private key (protected using the public key of the other) for possible re-assembly of nodes.

Public key generation and issuance

The certified package contains the CA verified node's public key certification named DC, the node's identity, the CA's identity, and the certificate's expiry period. In addition, it contains the certificate's expiry deadline, and the host must renew the pair of keys when the terminating period expires. As a result, all network nodes have set termination duration. The Digital Signature Algorithm (DSA) is employed solely to create DC, and DC is the shared key primitive of data integrity [35–37]. A DC is a method of binding an identity to digital content. The following is a summary of the certificate generating process using the DSA method:

Algorithm 3

```
Step 1: Choose an arbitrary value k between 0<k<q
Step 2: Create a hash code called H.
Step 3: Construct the Key 'K' using K = ($g^k$ mod p) mod q.
Step 4: DC = (H+K * Ti) mod q generates the Certificate DC.
Step 5: Save the DC in the {K,n} format.
Step 6: Eq. (6) is used to predict the distribution of public keys.
```

$$\check{\mathrm{T}}^{\mathrm{ᵇ}}_{\mathrm{i}\rightarrow \mathrm{j}}(t) > \check{\mathrm{T}}\mathrm{m} \tag{6}$$

If the preceding threshold is satisfied for node i and its neighboring node j, the node continues to spread the key to its trusted 1-hop neighbor. If the trust rate of neighbor j goes underneath the minimum trust level of node i, the node does not transfer the key to a neighboring node. The entire investigation is carried out based on the trust level, ensuring that the keys always are retained safe, eliminating key leaks, and compromising keys.

4 Simulation Performance and Analysis

In the presence of vulnerable nodes environment, we use the Network Simulator (NS) 2.34 [38] to experiment with the HANT protocol. First, we compare the proposed scheme HANT's simulated outcomes to the NB-DPC [21] and Gautam et al. [20] protocols. Then, we use the following parameters to assess HANT's efficacy: Packet Delivery Ratio (PDR), Detection Ratio (DR), Average end-to-end (E2E) latency, Network Life Time (NLT), and False Positive Rate (FPR). Tab. 2 lists the various factors used during modeling. For example, for a fixed beginning trust value of 5.0, the minimum needed trust threshold is 3.0 in experiments.

Table 2: Notations and meaning

Parameter	Values
Simulation tool	NS-2.34
Simulation area	750 m x 750 m
Total simulation time	550 s
Total nodes	100
Transmission range	250 m
Traffic type	CBR (UDP)
No. of connections	25
Movement model	Random waypoint
CBR rate	0.2 Mbps

(Continued)

Table 2 (continued)

Parameter	Values
Maximum speed	22 m/s
Maximum number of vulnerable nodes	5
Pause time	0, 10, 15, 25 s
Constant (w)	4 (range 0–1)
Trustworthy index threshold	0
Initial energy	350 J
Detect_Suspect interval	8 s
rx Power	1 W
idlePower	1 W
tx Power	1 W
Packet size	512 bytes

4.1 Throughput

Tab. 3 shows the performance analysis between the HANT and conventional technologies, depicted in Fig. 3. The active sensor nodes in our proposed HANT remain operational despite the emergence of vulnerable nodes since the trustworthiness assessment is arbitrary for a 1-hop neighbor examination.

Table 3: Throughput

Vulnerable node	HANT	NB-DPC	Carlier et al.
2	180	160	158
4	177	155	163
6	174	150	155
8	170	140	144
10	165	130	135

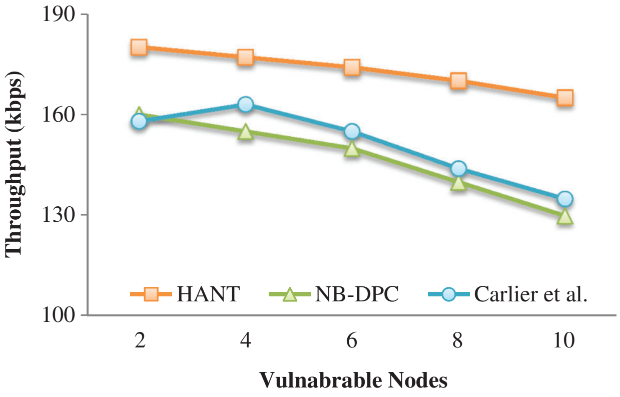

Figure 3: Throughput

Furthermore, vulnerable nodes are dynamically banned from the routing process to maintain communication throughout the timeframe [39]. To route signals to the BS, the CA chooses nodes that have been stable throughout time. As a consequence, the nodes measure overall for continuous communication, with the channel's performance remaining 40.45 percent greater and 35.5 percent better than the previous NB-DPC and Carlier et al. models.

4.2 Energy Savings

Tab. 4 and Fig. 4 depict compromised nodes' impact on power consumption. The CA has a lesser effect, with highly dynamic decisions and restricted measure assessment in an unexpected way instead of challenging a node to make critical judgments sequentially. As a result, in the presence of vulnerable nodes, minimal energy is consumed on transmissions because the proportion of trusted nodes meeting the energy limit is significant.

Table 4: Energy saving (J)

Vulnerable node	HANT	NB-DPC	Carlier et al.
2	5	8	12
4	8	12	15
6	10	19	24
8	14	22	26
10	18	25	30

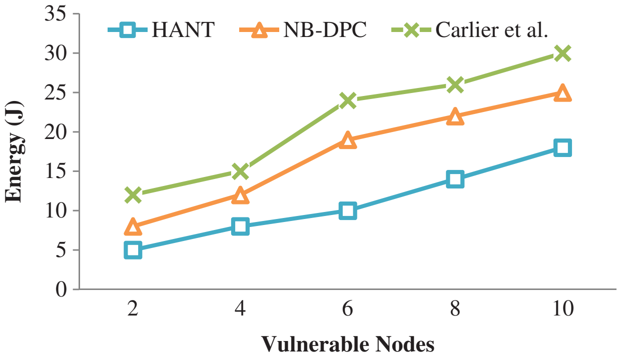

Figure 4: Energy saving

Furthermore, it prohibits conscious choices for the same transmitting operation made by cluster analysis. As a result, the HANT outperforms the NB-DPC and Carlier et al. in energy saving by 44.53 and 26.66 percent, respectively.

4.3 Average E2E Latency

When compared to conventional techniques, E2E delay is the statistic used to evaluate, and it clearly states how quickly the identification of vulnerable nodes and the calculated value records in Tab. 5. The visual depiction of E2E latency depicts in Fig. 5 based on Tab. 5.

Table 5: Average E2E latency (ms)

Vulnerable node	HANT	NB-DPC	Carlier et al.
2	2.51	4.35	3.52
4	3.62	5.37	4.5
6	4.43	8.53	6.43
8	7.58	14.12	10.5
10	8.42	20.5	13.62

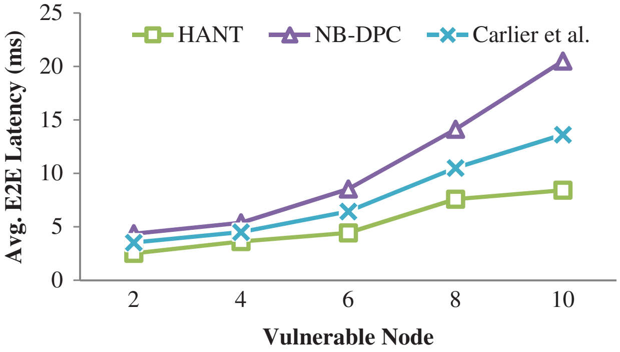

Figure 5: Avg. E2E latency

Due to repetitive trustworthiness and power testing procedure, the number of vulnerable nodes increases network delay. In addition, the reliability, power updating, and rebalancing parameters that occur once to detect a fraudulent node add to the network latency.

Our HANT approach will not regularly perform trust and power evaluation; instead, the arbitrary function uses restricted verification parameters to designate a vulnerable node [40–42]. This procedure can be quickened for a local update, reducing the time required to stop communications. As a result, the network's total delay has decreased. Compared to the existing approach, the HANT reduces the latency by 62.5 and 53.6 percent, respectively.

4.4 Network Lifetime

The influence of vulnerable nodes gradually exhausts network life, resulting in connection termination. In Tab. 6, The HANT preserves the nodes' power through a random function by monitoring their actions in exceeding vulnerable node influence. The lifespan of the entire network extends because it preserves the power. For example, when the vulnerable nodes are 10 in numbers, the NL in NB-DPC approach is 280 s, and the Carlier et al. model is 300 s, as illustrated in Fig. 6, whereas the saved lifetime in our suggested HANT is 375 s.

Table 6: Network lifetime (s)

Vulnerable node	HANT	NB-DPC	Carlier et al.
2	475	450	420
4	450	425	385
6	425	400	350
8	400	375	330
10	375	300	280

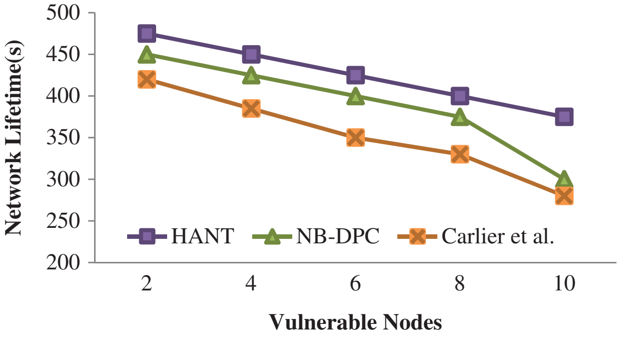

Figure 6: Avg. E2E latency

4.5 Detection Rate

The detection rate across the trustworthiness update period depict in Fig. 7. The volume and frequency of level concerning the neighbor's trust reduce as the trustworthiness update period grows, limiting the detection of such nodes as in Tab. 7. Two steps of node trust assessment are used in HANT: direct trustworthiness that satisfies power restrictions and an additional measure analysis.

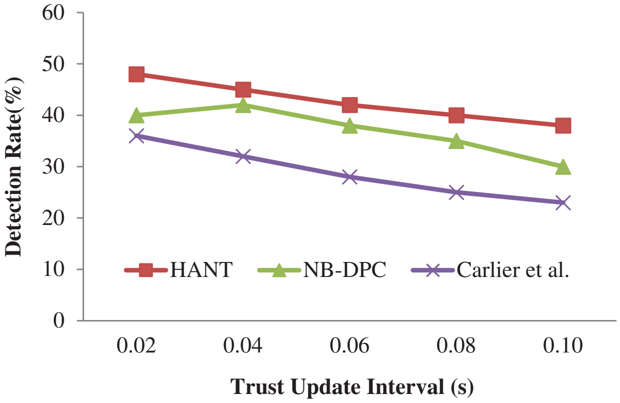

Figure 7: Detection rate (%)

Both efforts focus on detecting and separating vulnerable nodes from the routing process to minimize their network effects. The detection is continual through the cross-examination functionality of path trustworthiness, interaction quality, and responsiveness. For example, when the trustworthiness updating period is 0.1 s, our suggested HANT finds 35% of susceptible nodes.

Table 7: Detection rate (%)

Trust update interval	HANT	NB-DPC	Carlier et al.
0.02	48	40	36
0.04	45	42	32
0.06	42	38	28
0.08	40	35	25
0.10	38	30	23

4.6 False Positive Rate

The FPR is due to differences in trust updating intervals, as seen in Fig. 8 and Tab. 8. As the trust updating interval changes, it becomes more challenging to recognize trustworthy nodes, leading to higher levels of false-negative in the system. False negatives use to decrease the number of FPR in the arrangement. The incremental metric assessment of node based on its relationship quality and response time can prevent the propagation of false-negative in the HANT. The response time is the most basic metric to identify a suspicious node and leads to significant verification. It is a continuous procedure that takes more time; however, in the HANT, an alternative is appointed to continue the routing process. False-negative reduces when messages are sent diligently via trusted nodes, lowering FPR. FPR is reduced by 1.43 percent more with the HANT than earlier models.

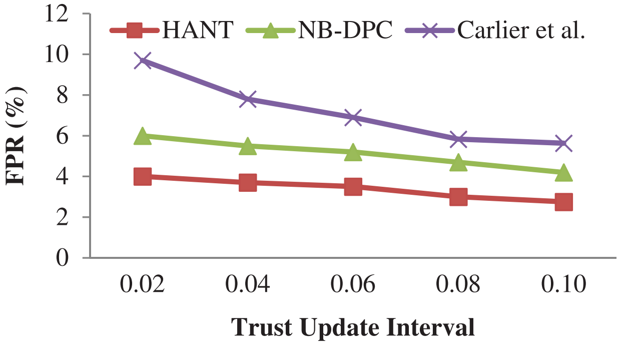

Figure 8: False positive rate (%)

Table 8: False positive rate (%)

Trust update interval	HANT	NB-DPC	Carlier et al.
0.02	4	6	9.7
0.04	3.7	5.5	7.8
0.06	3.5	5.2	6.9
0.08	3	4.7	5.83
0.10	2.75	4.2	5.63

5 Conclusions

In our study, we created a hybrid authentication utilizing node trustworthy for detecting Vulnerable Nodes in *Ad hoc* Sensor Network strategy for cluster-based ASN. First, the nodes are grouped and calculate the trust level of every node. The trust level is the combined effect of primary (direct) and secondary (indirect trust) and the key generation mechanism. Then, the cluster heads (CH) are chosen based on the predicted trust level. After that, a group of CAs checks up on every node. The transmitter then authenticates the communications and encodes using a key pair issued by a certificate authority, which the recipient then decodes. Finally, it authenticates the technique by verifying the identity of the sender and recipient. The suggested methodology reduces the E2E delay, NL, and energy-saving and increases other performance measures according to simulation findings. We concentrate further on our research to develop hybrid robust encryption strategies.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] V. Geetha and K. Chandrasekaran, “Enhanced beta trust model for identifying insider attacks in wireless sensor networks,” *International Journal of Computer Science and Network Security*, vol. 13, no. 8, pp. 14–20, 2013.

[3] F. Ishmanov, A. S. Malik, S. W. Kim and B. Begalov, “Trust management system in wireless sensor networks: Design considerations and research challenges,” *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 2, pp. 107–130, 2015.

[4] F. Bao, I. Chen, M. Chang and J. Cho, “Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection,” *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 169–183, 2012.

[5] S. Ganeriwal, L. K. Balzano and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” *ACM Transactions on Sensor Networks*, vol. 4, no. 3, pp. 1–37, 2008.

[6] Z. Yao, D. Kim and Y. Doh, “PLUS: Parameterized and localized trust management scheme for sensor networks security,” in *IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems*, Vancouver, BC, Canada, pp. 437–446, 2006.

[7] J. Duan, D. Yang, H. Zhu, S. Zhang and J. Zhao, “TSRF: A trust-aware secure routing framework in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, pp. 1–14, 2014.

[8] J. Lopez, R. Roman, I. Agudo and C. Fernandez-Gago, “Trust management systems for wireless sensor networks: Best practices,” *Computer Communications*, vol. 33, no. 9, pp. 0140–3664, 2010.

[9] H. Chen, H. Wu, J. Hu and C. Gao, “Event-based trust framework model in wireless sensor networks,” in *2008 Int. Conf. on Networking, Architecture, and Storage*, Chongqing, China, pp. 359–364, 2008.

[10] A. Perrig and J. D. Tygar, “TESLA broadcast authentication,” in *Secure Broadcast Communication*, Boston, Springer, pp. 29–53, 2003.

[11] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, “SPINS: Security protocols for sensor networks,” *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[12] B. C. C. Lai, D. D. Hwang, S. P. Kim and I. Verbauwhede, “Reducing radio energy consumption of key management protocols for wireless sensor networks,” in *Proc. of the 2004 Int. Sym. on Low Power Electronics and Design (IEEE Cat. No.04TH8758)*, Newport Beach, CA, USA, pp. 351–356, 2004.

[13] S. Zhu, S. Setia, S. Jajodia, LEAP, “Efficient security mechanisms for large-scale distributed sensor networks,” in *Proc. of the 10th ACM Conf. on Computer and Communications Security (CCS ‘03). Association for Computing Machinery*, New York, NY, USA, pp. 62–72, 2003.

[14] Y. Zhang, Y. Shen and S. Lee, "A cluster-based group key management scheme for wireless sensor networks," in *12th Int. Asia-Pacific Web Conf.*, Busan, Korea (South), pp. 386–388, 2010.

[15] L. Li and X. Wang, "A high security dynamic secret key management scheme for wireless sensor networks," in *Third Int. Sym. on Intelligent Information Technology and Security Informatics*, Jian, China, pp. 507–510, 2010.

[16] A. S. Elqusy, S. E. Essa and A. El-Sayed, "A key management techniques in wireless sensor networks," *Communications on Applied Electronics*, vol. 7, no. 2, pp. 8–18, 2017.

[17] K. Ren, W. Lou, B. Zhu and S. Jajodia, "Secure and efficient multicast in wireless sensor networks allowing ad hoc group formation," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 2018–2029, 2009.

[18] M. Carlier, K. Steenhaut and A. Braeken, "Symmetric-key based security for multicast communication in wireless sensor networks," in *2018 4th Int. Conf. on Cloud Computing Technologies and Applications (Cloudtech)*, Brussels, Belgium, pp. 1–6, 2018.

[19] J. Ding, H. Zhang, Z. Guo and Y. Wu, "The DPC-based scheme for detecting selective forwarding in clustered wireless sensor networks," *in IEEE Access*, vol. 9, pp. 20954–20967, 2021.

[20] A. K. Gautam and R. Kumar, "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks," *SN Applied Sciences*, vol. 3, no. 50, pp. 393, 2021.

[21] J. W. Ho, M. Wright and S. K. Das, "Distributed detection of mobile malicious node attacks in wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 512–523, 2012.

[22] S. M. Wa Umba, A. M. Abu-Mahfouz, T. D. Ramotsoela and G. P. Hancke, "A review of artificial intelligence based intrusion detection for software-defined wireless sensor networks," in *2019 IEEE 28th Int. Sym. on Industrial Electronics (ISIE)*, Vancouver, BC, Canada, pp. 1277–1282, 2019.

[23] T. T.Vandervelden, R. D. Smet, K. Steenhaut and A. Braeken, "Symmetric-key-based authentication among the nodes in a wireless sensor and actuator network," *Sensors*, vol. 22, no. 4, pp. 1403(1-11), 2022.

[24] D. Liu and P. Ning, "Multilevel µTESLA: Broadcast authentication for distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 4, pp. 800–836, 2004.

[25] D. Liu, P. Ning, S. Zhu and S. Jajodia, "Practical broadcast authentication in sensor networks," in *The Second Annual Int. Conf. on Mobile and Ubiquitous Systems: Networking and Services*, San Diego, CA, USA, pp. 118–129, 2005.

[26] M. Nakkar, R. Altawy and A. Youssef, "Lightweight broadcast authentication protocol for edge-based applications," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11766–11777, 2020.

[27] G. Bertoni, J. Daemen, M. Peeters and G. V. Assche, "Sponge-based pseudo-random number generators," in *Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science*, S. Mangard, FX. Standaert (eds.), vol. 6225. Berlin, Heidelberg: Springer, pp. 33–47, 2010.

[28] D. Dhinakaran and P. M. Joe Prathap, "Preserving data confidentiality in association rule mining using data share allocator algorithm," *Intelligent Automation & Soft Computing*, vol. 33, no. 3, pp. 1877–1892, 2022.

[29] N. Partheeban, K. Sudharson and P. J. Sathish Kumar, "SPEC- serial property based encryption for cloud," *International Journal of Pharmacy & Technology*, vol. 8, no. 4, pp. 23702–23710, 2016.

[30] K. Sudharson, A. Mudassar Ali and N. Partheeban, "NUITECH – natural user interface technique formulating computer hardware," *International Journal of Pharmacy & Technology*, vol. 8, no. 4, pp. 23598–23606, 2016.

[31] D. Dhinakaran and P. M. Joe Prathap, "Ensuring privacy of data and mined results of data possessor in collaborative ARM," in *Pervasive Computing and Social Networking. Lecture Notes in Networks and Systems*. vol. 317. Singapore: Springer, 2022.

[32] S. Arun and K. Sudharson, "DEFECT: Discover and eradicate fool around node in emergency network using combinatorial techniques," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 10, pp. 1–12, 2020.

[33] K. Sudharson and V. Parthipan, " A Survey on ATTACK – Anti terrorism technique for adhoc using clustering and knowledge extraction," in *Advances in Computer Science and Information Technology. Computer Science and Engineering. CCSIT 2012. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*. vol. 85. Berlin, Heidelberg: Springer, pp. 508–514, 2012.

[34] A. A. Abins, J. Katiravan and S. M. Udhaya Sankar, “Performance optimization using heuristic approach in opportunistic WSR,” *Dynamic Systems and Applications*, vol. 30, no. 8, pp. 1304–1317, 2021.

[35] K. Kowshika, M. Ramakrishnan, J. Raja and S. M. Udhaya Sankar, “Energy aware detection and prevention of packet drop attack in wireless and mobile adhoc networks by packet drop battling mechanism,” *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 10, pp. 1882–1892, 2020.

[36] T. Sujithra, M. Sumathi, M. Ramakrishnan and S. M. Udhaya Sankar, “ID based adaptive-key signcryption for data security in cloud environment,” *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 4, pp. 205–220, 2020.

[37] S. M. Udhaya Sankar, V. Vijaya Chamundeeswari and K. Jeevaa, “An enhanced method to detect and prevent wormhole attach in m commerce,” *Asian Journal of Information Technology*, vol. 16, no. 1, pp. 77–81, 2017.

[38] S. M. Udhaya Sankar and V. Vijaya Chamundeeswar, “JIGSPASSZLE: A novel jigsaw based password system using mouse drag dynamics,” *Middle-East Journal of Scientific Research*, vol. 21, no. 11, pp. 2039–2051, 2014.

[39] S. M. Udhaya Sankar, V. Vijaya Chamundeeswari and K. Jeevaa, “Identity based attack detection and manifold adversaries localization in wireless networks,” *Journal of Theoretical and Applied Information Technology*, vol. 67, no. 2, pp. 513–518, 2014.

[40] D. Dhinakaran, P. M. Joe Prathap, D. Selvaraj, D. Arul Kumar and B. Murugeshwari, “Mining privacy-preserving association rules based on parallel processing in cloud computing,” *International Journal of Engineering Trends and Technology*, vol. 70, no. 3, pp. 284–294, 2022.

[41] D. Dhinakaran, D. A. Kumar, S. Dinesh, D. Selvaraj and K. Srikanth, “Recommendation System for Research Studies Based on GCR,” in *2022 Int. Mobile and Embedded Technology Conf. (MECON)*, Noida, India, pp. 61–65, 2022.

[42] K. Sudharson, M. Akshaya, M. Lokeswari and K. Gopika, “Secure Authentication scheme using CEEK technique for Trusted Environment,” in *2022 Int. Mobile and Embedded Technology Conf. (MECON)*, Noida, India, pp. 66–71, 2022.