Research Article

A Secure Data Dissemination in a DHT-Based Routing Paradigm for Wireless Ad Hoc Network

Rukhsana Kousar,¹ Majed Alhaisoni,² Shahbaz Abid Akhtar ,¹ Nadir Shah,¹ Aamir Qamar,¹ and Ahmad Karim³

¹Computer Science Department, COMSATS University Islamabad, 44000, Pakistan

²Computer Science Department, University of Ha'il, Saudi Arabia

³Institute of Computing Department, Bahauddin Zakariya University, Multan, 66000, Pakistan

Correspondence should be addressed to Shahbaz Abid Akhtar; shahbaz.akhtar.abid@gmail.com

Received 22 October 2019; Revised 14 May 2020; Accepted 25 May 2020; Published 24 August 2020

Academic Editor: Pierre-Martin Tardif

Copyright © 2020 Rukhsana Kousar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Over the past decade, distributed hash table- (DHT-) based routing protocols have been adopted in wireless ad hoc networks (WANETs) to achieve scalability in the route discovery phase by avoiding the flooding mechanism. The security aspects of the routing protocols based on the DHT mechanism are crucial to address and have not been discussed in the existing literature. Therefore, addressing the security issues in DHT-based routing protocols would prevent the service disruption, decrease the traffic overhead, and reduce the packet loss in the network. In this paper, several security issues are identified and elaborated through an example scenario. Moreover, a novel DHT-based routing protocol is proposed that uses a digital signature and the user's trust in order to swap securely the logical identifiers (LIDs). Trust between nodes is established by the user's acquaintance and the first visual contact. The proposed protocol vindicates its effectiveness via simulation results in terms of computation time, normalized overhead, percent improvement, and packet delivery ratio.

1. Introduction

Over the last two decades, wireless ad hoc networks (WANETs) and their subdomains such as wireless industrial networks, wireless sensor networks (WSNs), and vehicular ad hoc networks (VANETs) have been successfully implemented in different areas of life (e.g., military communications, emergency situations [1], airports, railway, subways, conferences, campuses [2], and intelligent transportation systems [3]).

Data routing and traffic control are the core of WANETs. Recent studies proved that the DHT-based routing protocols (e.g., 3D-RP [4–5], M-DART [6], VCP [7], Motion-MX [8], 3DCRP [9], and LA-3D-IS [10]) are better in performance compared to the traditional routing protocols, which use flooding in the route discovery phase (e.g., AODV [11], OLSR [12], and DYMO [13]). The DHT-based routing mechanism [14] avoids/reduces the network-wide flooding

in the route discovery phase, thus increasing the scalability in WANETs.

A universal identifier (UID) (i.e., MAC address or IP address) and logical identifier (LID) are used to identify a node in DHT-based routing protocols. Each node sustains its disjointed LID space portion (LSP) of the whole logical space (LS). Each node calculates its LID from predefined LS on the basis of its physical neighboring nodes. Furthermore, a logical identifier structure (LIS) for each node (e.g., ring, chord, or multidimensional structure) retains the path of its 1-hop/2-hop logical neighboring nodes (L_{nbr}). Therefore, a logical network is built on the top of a physical network. Each node uses its LID to forward the data packets/control packets in the network. Figure 1 shows the basic working of DHT-based addressing, lookup, and routing mechanisms. A network is established according to 3D routing [4, 5, 15], and the LIDs of nodes are in an ordered three-tuple $\{x|y|z\}$ -dim, where x , y , and z are values within the range of 0 to ± 1024 .

FIGURE 1: Illustration the following process: (i) AN computation; (ii) greedy LID-based routing.

Each node in a DHT-based routing protocol stores its LID and other mapping information like UID (IP/MAC addresses) at its *anchor node* (AN). For this purpose, a hashed function is applied on UID of each node for generating a hashed value $h(v)$ to find its AN that is drawn from the LS according to the protocol specification. Node y finds its AN that is node x having LID closest to the $h(v)$ value. To send the data packet to a destination node q , a source node s first retrieves the LID of the destination node q from its AN, say p , whose LID is obtained by applying the same hash function at the UID of node q . The node s then forwards a lookup message towards node p that is obtained from the hash value to get the destination node q 's LID. Upon receiving q 's LID, the source node s sends the data message to destination node q based on the LID of node q . To forward the data message, each node compares the node q 's LID with its own and the LIDs of its 1-hop/2-hop L_{nbr} . If the LID of destination node q is nearest to its own LID, the node itself is the destination node; otherwise, the node forwards the message to a logical neighbor that has the LID nearest to that of the node q . The important terms related to DHT-based secure routing in WANETs is shown in Table 1.

For instance, node l stores the mapping information of node f as shown in Figure 1. To communicate with destination node f in the network, node m simply applies the hash function over the UID of node f and gets a hashed value that refers to node l 's LID, that is, AN of node f as shown in Figure 1. Node m then sends a LID lookup request message (MREQ) containing m (LID, UID) and f (UID, hashed value) towards node l to get node f 's LID.

The path followed by MREQ in the network is shown in Figure 1. Now, node l sends a mapping reply message (MREP) to node m for direct communication. After getting the LID of node f , the source node m directly communicates with destination node f by following the aforementioned method.

In this paper, DHT-based routing protocols are considered that are adopted in the MANETs to achieve scalability by avoiding flooding in the route discovery phase. Addressing the security of a routing protocol is crucial to enhance the reliability of the network [16–18]. In the past, researchers have proposed various secure versions of the traditional routing protocols, e.g., S-AODV [19], SOLSR [20], and S-DYMO [21]; a survey of these approaches can be found in [22–24]. Node addressing, lookup, and routing mechanisms in the DHT-based routing protocols are different from the traditional routings, which raise new security issues that may degrade the performance and reliability of the network. In this paper, we highlight various security issues associated with the DHT-based routing protocols, and addressing those would prevent service disruption, decrease the traffic overhead, and reduce the packet loss in the network.

In addition, a secure DHT-based routing protocol for WANETs is proposed that uses the digital signature and user's trust for the node authentication and sophisticated encryption/decryption algorithm to securely transfer the data/control packets among the nodes. To the best of our knowledge, none of the existing literature has debated the security aspects in the context of the DHT-based routings at the network layer, and none has proposed the secure version of any of these protocols.

TABLE 1: Important terms related to DHT-based secure routing in WANETs.

Symbol	Description
AN	<i>Anchor node(AN)</i> stores the mapping information of other nodes with respect to its LSP
LID	Logical identifier of a node
LS	Logical identifier space of node
LSP	LS portion of a node
UID	Universal identifier of a node
T_{ix}	Tuple of node i in x -axis
T_{iy}	Tuple of node i in y -axis
T_{iz}	Tuple of node i in z -axis
LSP_{ix^+}	LS portion of node i in positive x -axis
LSP_{ix^-}	LS portion of node i in negative x -axis
LSP_{iy^+}	LS portion of node i in positive y -axis
LSP_{iy^-}	LS portion of node i in negative y -axis
LSP_{iz^+}	LS portion of node i in positive z -axis
LSP_{iz^-}	LS portion of node i in negative z -axis
T_{kx}, T_{ky}, T_{kz}	Three tuples of node k in x -, y -, and z -axis
W_{mk}	Joining node m assigned weight to its neighbor node k
W_{mj}	Joining node m assigned weight to its neighbor node k
L_{nbr}	1-hop logical neighbors of a node
Sig_n	User signature
PI	Personal identity of node
PbK	The public key of node
IP	IP address of node
PrK	The private key of node
IC	Identity card of node
C	Certificate of node
Nbr_{List}	Neighbor list: a list containing LIDs of neighboring node and neighbors of the neighboring nodes
SD	Sum of difference
LSD	Least sum of difference
T_{dx}, T_{dy}, T_{dz}	Three tuples of hashed value or destination's LID
$T_{nbrx}, T_{nbry}, T_{nbrz}$	Three tuples of neighbor LID
$h(v)$	Hash value
SII	Source index information

Our main contribution in this paper is

- (1) addressing the security issues related to DHT-based routing protocols
- (2) identifying possible security vulnerabilities from the attacker's end
 - (a) destination LID tempering attack
 - (b) false neighborhood attack

- (i) LID computation at the joining node
- (ii) LID assignment by the neighboring node
- (c) authentication of the node's LID
- (3) devising a novel DHT-based routing protocol to mitigate the threats
 - (a) this protocol uses the digital signature and user's trust in order to swap securely the LIDs
 - (b) trust between nodes is established by a user's acquaintance and the first visual contact

In DHT-based routing protocols, the LID computation of a newly joining node is based on the neighboring nodes, and the same is used for the routing of data from the source to the destination node. In case if the neighboring node(s) is not legitimate, the joining node would result in the computation of the falsified LID regarding its physical proximity. Moreover, the same LID is used later in lookup and routing processes that leads to longer routes, increased delay, and packet loss or may completely halt the communication of the DHT-based routing protocol. Initially, we have specifically focused on addressing the security threats discussed in Section 2. The proposed security mechanism has not only successfully addressed the security threats elaborated in Section 2 but also proven to be effective and efficient through simulation.

The rest of the paper is organized as follows: Section 2 identifies and discusses several security vulnerabilities in the DHT-based routing protocols, Section 3 provides a brief description of the existing protocols, Section 4 represents a proposed novel secure DHT-based routing protocol, Section 5 presents an evaluation of proposed protocol by formal methods, and Section 6 discusses simulation results. The paper conclusion and future work have been discussed in Section 7.

2. Security Vulnerabilities in Context of DHT-Based Routing Protocols

The existing protocols that employ the DHT mechanism for routing at the network layer are insecure and vulnerable to several potential security threats. In this section, we identify possible security vulnerabilities, from the attacker's end in the routing protocols that are using the DHT mechanism, and describe the prerequisites to mitigate these threats. The handling of security vulnerabilities for DHT-based routing protocols is crucial for their smooth functioning and might disrupt entirely the communication among nodes.

Two types of nodes can cause service disruption in the network: (i) malicious node and (ii) compromised node. A node is referred to as malicious if it cannot authenticate itself and prove its legitimacy due to invalid cryptographic information, while a node is referred to as compromised if it is a legitimate node that can authenticate itself and is trusted by the other nodes in the network but is behaving maliciously. We use the term attacker for both malicious and

compromised nodes. The following are the attacks that can be initiated by malicious or compromised nodes in DHT-based routing protocols.

2.1. Destination LID Tampering Attack. For a LID lookup message, an attacker can send a modified or generate a LID lookup reply with falsified information by following two ways: (i) An attacker may modify the content of the LID lookup reply message and return to the source with fabricated information. (ii) After receiving a LID lookup request, the attacker as an intermediate node generates the LID lookup reply with falsified information, instead of forwarding the lookup request to the actual AN. More specifically, these scenarios are explained through an example as follows:

The node f stores its mapping information at node l as shown in Figure 2. To communicate with node f , node m sends a LID lookup request message (MREQ) containing m (LID, UID) and f (UID, hashed value) towards node l to get node f 's LID. The path followed by MREQ in the network is shown in Figure 2. For instance, if an intermediate node p is malicious or compromised, it can easily disrupt the communication by generating a mapping reply message (MREP) for node m and adds its LID (p 's LID) along with the UID of node f in MREP. Node p then sends the MREP towards node m . Similarly, suppose the MREQ arrives at the AN l . Node l generates MREP by adding node f 's LID and sends MREP towards node m . After receiving MREP, an attacker p can modify the MREP by replacing f 's LID with its own LID (p 's LID) and forwards the modified MREP towards node m . In either case, upon receiving the MREP, node m would take the LID of node p in MREP as the LID of the destination node f , and thus, the data will be sent to node p rather than to node f as shown in Figure 2. In this way, by tampering with the destination node f 's LID with its own LID, node p would receive all the data packets that are destined for node f . This attack is crucial to address because AN is used in all DHT-based routing protocols to achieve communication among nodes.

By tampering with the LIDs in the MREP, it could result in loss/theft of information that is critical for the communication in the network.

2.2. False Neighborhood Attack. In the routing protocol using the DHT mechanism, LID is allotted to the newly joined node depending on its one-hop neighboring node's LID [4, 5, 7]. Thus, the newly joining node in the network obtains the LID from its one-hop neighboring nodes. The purpose of assigning the LID to a node is twofold: (i) to arrange nodes in the LIS over the physical topology (PT) with respect to their LIDs; (ii) to route the data/control packets among the nodes of the network.

2.2.1. LID Computation at the Joining Node. In this case, the joining node gets 1-hop/2-hop neighbor information, i.e., their LIDs, and computes its LID, i.e., its relative position, by using its neighbors' LIDs as proposed in [4, 5, 7]. It is crucial for the joining node to compute an accurate LID as it reflects the physical proximity of the joining node with

respect to its logical neighbors in the logical network. Thus, the accurate LID of physical neighboring nodes is crucial for computing the accurate joining node's LID, which will later be used for lookup and routing of data/control packets. In case a neighboring node is not a legitimate neighbor and the joining node obtains a fake/wrong LID of its neighboring node through a hello message, then it would result in the computation of an incorrect LID of the joining node with reference to its physical proximity. This may lead to the mismatch problem [4, 25] which exaggerates the end-to-end delay and routing overhead that would eventually affect the performance of the DHT-based routing protocol in terms of its reliability. This is a crucial security attack and must be addressed for the smooth functioning of DHT-based routing protocols.

2.2.2. LID Assignment by the Neighbor Node. In this case, a newly joining node acquires the LID from its one-hop neighboring nodes [25]. If the joining node gets nearer to a neighboring node that is a malicious or a compromised node, then the joining node obtains a fake/wrong LID that would not interpret its intraneighbor relationships with its neighboring nodes in the LIS. The compromised neighboring node may assign a falsified LID without considering its LSP, which may affect the forwarding of both the data and control packets. In any DHT-based routing protocol, it is a crucial security attack and must be handled carefully.

2.3. Authentication of a Node's LID. The existing DHT-based routing protocols cannot verify whether a particular LID belongs to the node or not. For example, in Figure 2, suppose node p stores its own LID $\{256|1|1\}-1$ and UID at AN l pretending that LID $\{256|1|1\}-1$ belongs to node f . Thus, the anchor node l is unable to verify through any mechanism whether it is node f 's LID or not. In this case, if m is a source node and needs to communicate with node f , then node m requires the LID of destination node f from its AN l . Node l would return the LID of node p to source node m , which would send the data packet to node p .

Although, the above security attacks have been discussed for traditional/non-DHT-based routing protocols, and researchers have given various possible solutions to the above issues. The routing protocols that are using the DHT mechanism cannot deploy those solutions directly at the logical network because their functionality of the DHT-based routing protocol is different from the traditional routing protocol. Traditional routing protocols mainly rely on IP addresses for routing of packets. The above security threats may completely halt the functionality of routing protocols that are using DHT mechanism, and as a result, total communication disruption may occur among nodes. The solution to the above security threats is crucial for the smooth functioning of DHT-based routings. To the best of our knowledge, none of the existing literature provides a secure version of the routing protocol using the DHT mechanism that works at the network layer. This paper is the first to attempt the solution of such security threats in the context of DHT-based routing protocols.

established based on the eye gaze contact. Moreover, they have used the node's LID and IP to introduce name service that is distributed.

Zhan et al. [26] state that wireless channels cause problems in key generation and key distribution, e.g., the shared secret key encoded bit sequence suffers from an extremely high bit mismatch rate. The authors propose an efficient key generation scheme by using the curve fitting technique to preprocess the channel measurements so that original channel measurements are better during reciprocity.

Filipek and Hudec [27] suggest a secure architecture that is based on a trust model along with PKI, intrusion prevention system (IPS), and firewall for the distributed environment in mobile ad hoc networks (MANETs). Nodes' trust level defines their privileges L0 to L3 that can be reduced or revoked upon malicious behavior. L0-level node has only the privilege to request the certificate. L1-level nodes are allowed for the end-to-end communication in the network. L2-level nodes can participate in the routing, IPS, and distribution storage. L3-level node is a stand-alone attribute authority (AA) that can certify to other nodes and create its own ecosystem. The certificate is signed by AAs, and every node can verify its validity by applying the AA public key. In the proposed architecture, a firewall serves as security overlay and PKI brings confidentiality and data communication policy enforcement, whereas IPS is essential for controlling nodes and makes sure they follow security policies (PKI and firewall rules). However, the authors do not consider DHT-based routing.

Xia et al. [28] propose a lightweight trust-enhanced ad hoc on-demand multipath distance vector protocol (TeAOMDV) that is an extension of ad hoc on-demand multipath distance vector protocol (AOMDV). This trust framework provides a choice of an optimal two-way trusted route that mitigates the impairment effect from such entities. The monitoring entity collects the passive and local information via a promiscuous mode that is used to evaluate the behavior of interested entities to translate an estimate of the trust. Passive acknowledgment uses the promiscuous mode to monitor the neighbor's behavior. Local information means the node's local memory stores the satisfactory evaluation between two neighboring nodes. The proposed technique used a trusted approach into a source routing mechanism. The sending node evaluates the routing path before forwarding the data using features like node reputation or identity information. After satisfactory evaluation, the sender node forwards the packet and stores it in a packet buffer; the senders monitor the packet forwarding in the promiscuous mode. After detecting the successful packet forwarding, the corresponding correct forwarding is increased by 1 and is removed from the packet buffer. This technique reduces the route discovery frequency and routing overhead. However, it would not perform well in a DHT-based routing because the traditional routing (e.g., AODV) is different than a DHT-based routing.

Sathiya and Gomathy [29] state that an intermediate node as an attacker along the path from the source to the des-

tination can interrupt the data access in MANETs. They propose a new solution by using the Beer-Quiche theoretical routing model [32] in which the source node tracks the available path at each stage, the path enduring bandwidth and the attacker policy, that collects the information made available by the previous stage. The source node selects an optimal path for packet broadcasting based on this information. Moreover, they suggest a proper switching mechanism to select multiple paths from the source to destination. Again, they do not consider the DHT-based routing scenario of WANETs.

Fu et al. [31] encounter a replication attack (RA) associated with random predistribution key approaches (RPK). The paper contribution is fivefold. First, the replication attack is modeled; second, the flexibility/operability of the RPK is measured; third, it analyzes, characterizes, and discusses the relationship among the cloned node; fourth, it further evaluates and compares through experiments the efficiency of several approaches against RA; and fifth, it anticipates the consequences that an attacker can obtain by introducing multiple malicious/compromised nodes in the network.

Rajkumar and Narsimha et al. [33] state that the public key certificate is used in MANETs to establish trust between two communicating nodes. So, to augment the network security and reduce attacks generated by the network nodes, it needs to escort an effective mechanism for validation and certificate revocation. The authors proposed a new mechanism that revokes and distributes certificates for a node based on a threshold value which is computed on the trust level. The trust value is based on direct trust (T) and indirect trust (T') values. T value is computed by

$$T_{xy} = T_{xy}(pr) * \vartheta + RSI_{xy} * (1 - \vartheta), \quad (1)$$

where the previous direct trust value of a node is mentioned by $T_{xy}(pr)$, node N_y in node N_x to the inclusion of a recent satisfaction index (RSI_{xy}) with direct neighbor nodes, and ϑ shows a constant value. Each node periodically computes its RSI value by

$$RSI_{xy} = \%s(x, y) - \%l(x, y), \quad (2)$$

where $\%s(x, y)$ shows that node N_x initiated a percentage of packets that is forwarded by node N_y over the total number of packets provided to N_y and $\%(l)(x, y)$ shows that the percentage of expired packets over the total number of packets given to N_x is a constant value that shows the confidence level of stored N_x for N_y . The indirect trust T' value is calculated as an aggregated trust report received and processed by N_x to N_y . It is computed by

$$T'_{xy} = \frac{\sum RT_{nx} * \rho n * RT_{nx}}{\sum \rho n + RT_{nx}}, \quad (3)$$

where pn is the degree centrality of the reporting node. Each node N_x calculates its centrality by using

$$C_x = \frac{1}{\sigma} \sum_{y \in w(i)} C_y = \frac{1}{\sigma} \sum_{y=1}^n V_{x,y} C_y, \quad (4)$$

where $w(i)$ is a set of i^{th} linked nodes, σ shows constant value, n is the sum of nodes and V_x , and j is the adjacency matrix of the network. After trust computation, there is a certificate authority (CA) that distributes the secret key to all nodes; thus, misbehaving nodes are eliminated. Moreover, the proposed technique also provides confidentiality, integrity, and secure multipath routing for data transmission in MANET. More specifically, a route request (RREQ) packet is signed with a digital signature for secure route discovery. When the destination receives the RREQ packet, it verifies all the signatures. Then, the destination node sends a route reply (RREP) through the same path. Finally, the path is accepted after verifying the signature by the source node. For secure transmission, a source node encrypts the fragmented message using a soft encryption, and data packets move via multiple paths to the destination. After receiving the encrypted message, the destination node decrypts the message to recover the original message. However, the proposed technique still faces the challenges on how to revoke and validate the certificate at a node.

Rajeswari et al. [34] proposed a secure routing approach that integrates two algorithms called a trust-based next forwarding node selection (TNFNS) algorithm and a fuzzy-based stable and secure routing (FBSSR) algorithm. The proposed mechanism provides a solution to the dynamic nature of MANETs with distinct characteristics like resource constraint, decentralized management, frequent mobility, dynamic topology, and control that leads to additional overhead in the provision of safe and steady routing. In order to enhance security during the routing process, trust values are used to isolate the malicious nodes. Only trusted nodes in the network with high residual energy and link stability perform a stable and secure routing of the network. Another contribution is FBSSR that enhances the AODV routing table with on-demand performance route discovery and route maintenance. The proposed algorithm can improve the overall performance and network lifetime using a fuzzy-based rule system to perform inference for selecting more secure and stable routes. It handles uncertainty in selecting the trusted nodes using qualitative analysis on trust values and link properties. The TNFNS algorithm is developed using network monitoring values and routing table values that increase reliability and data security in the communication process. Moreover, the proposed mechanism is capable of increasing the network performance compared to the related security algorithms.

Brindha et al. [35] proposed the fuzzy enhanced secure multicast routing (FSMR) scheme to make the data more secure in MANETs from active and passive attacks and ensure packet authentication and integrity. The proposed scheme uses an intelligent intrusion detection model to

observe the network and system to seek out intrusion activities. Anomaly-based IDS detects both network and computer intrusion and misuse by monitoring system activity and classifying it as either anomalous or normal. Certificateless routing with key generation, signcryption, and unsigncryption are used to authenticate the data and eliminate the intrusion in the network. In a key generation, initially, the main server generates all public parameters of the elliptic curve and publishes them. Each sensor node chooses a private key and calculates the related public key. The sender node performs signcryption; in signcryption, confidentiality and integrity are achieved in a single step that reduces the communication and computation cost and increases the efficiency, whereas encryption and signature need two steps. In unsigncryption, the receiving node decrypts the received signcryption text, extracts the plain text, and verifies the digital signature. The simulation result shows a better performance than existing schemes.

Arulkumaran and Gnanamurthy [36] present a solution for one of the possible attacks in MANETs that is the black hole attack. In the black hole attack, the malevolent node promotes itself such that it has the entire valid routes to its destinations. The proposed mechanism used fuzzy logic technique to detect a black hole attack. Fuzzy logic is mathematical logic in which the prediction values are assigned to an imprecise range of data to handle the problems. Fuzzy logic gives the certificate to only trusted nodes that help to identify misbehaving nodes. AODV uses a fuzzy-based trust model for packet route selection and avoids the black hole attack. Trust value is calculated using immediate neighbor trust values and recommendation trust values. The proposed mechanism gives less end-to-end delay, better throughput, and significant packet delivery ratio.

Liu et al. [37] proposed a security disjoint routing-based verified message (SDRVM) scheme to improve the network performance in terms of data arrival ratio, transmission delay, and consideration of the capacity for determining malevolent nodes and energy efficiency. SDRVM overcomes the energy efficiency issue of the sensor nodes that substantially affects the network security. The proposed scheme establishes two disconnecting dominating sets (a data connected dominating set (CDS) and verified message CDS) that are based on the remaining energy consumption among nodes. SDRVM adopts a method for recording ID information in data packets with an adjustable specified probability (marking as higher or smaller) according to the remaining energy of nodes and for logging ID information in nodes. The nodes include ID information into data packets with a specified probability when nodes send data packets to other nodes. If the node energy is insufficient, the ID information in data packets is locally stored/logged, and the marking probability is reduced. If the node has sufficient energy, the sensor node's recording probability is increased. The sensor node duty cycle will be increased to fully utilize the energy when the intensity of energy harvesting is strong. Otherwise, to save the energy, the sensor node duty cycle is reduced. Node higher duty cycle in the data CDS reduces the transmission delay. Node lower duty cycle in the v-message CDS saves energy.

TABLE 2: Summary of existing approaches.

Paper	Approach	Encryption algorithm	Key distribution	Central trust authority	Authentication	Confidentiality	Integrity	Hash function
[17]	Secure self-configured spontaneous network protocol	AES	RSA	Distributed CA	✓	✓	✓	SHA-1
[26]	IKM, an ID-based key management scheme	RSA	Threshold cryptography	Certificateless	✓	✓	✓	SHA-1
[27]	Secured routing protocol along with PKI, firewall, and IPS	AES	RSA	Centralized certification authority within an ecosystem	×	✓	✓	SHA-512
[28]	Trust-enhanced ad hoc on-demand multipath distance vector protocol (TeAOMDV)	×	×	Distributed trust-based mechanism not CA	✓	×	✓	MAC (message authentication code)
[33]	Trust-based certificate revocation for secure routing (TCRSR) protocol	Threshold cryptography	Threshold cryptography	Chain certificate	×	✓	✓	×
[35]	Fuzzy enhanced secure multicast routing (FSMR)	-	-	Certificateless	✓	✓	✓	✓
[36]	AODV with a fuzzy-based trust model	RSA	RSA	Centralized certificate authority	✓	✓	✓	MD4
[38]	Resistive to selective drop attack (RSDA)	ECC	-	-	✓	✓	✓	✓
[39]	Encrypted trust-based dolphin glowworm optimization (E-TDGO)	AES-128	-	Mutual trust relationship	×	✓	×	×
[31]	Random predistribution key approaches (RDK)	×	Random prekey distribution	-	-	×	-	-
Analytical modeling								
[29]	Stochastic Beer-Quiche multipath routing (SBQMR), a theoretical model	-	-	-	✓	-	-	-
Stochastic multipath routing mechanism is adopted based on Beer-Quiche theoretical model								
[34]	Fuzzy-based stable and secure routing (FBSSR)	-	-	-	-	-	-	-
Trust-based next forwarding node selection algorithm (TNFNS)+analytical analysis								

Poongodi et al. [38] proposed an effective lightweight security mechanism named resistive to selective drop attack (RSDA). In a selective drop attack, the neighboring nodes are not reliable in message forwarding to the next node. The identification of such nodes is crucial, and segregating them from the network is a challenging task. The RSDA technique detects malevolent nodes in the network under a particular drop attack that overloads a host and stops it from working. The throughput of a host may potentially drop to the minimum level. The elliptic curve digital signature algorithm is used to authenticate the nodes to accomplish reliable routing. The existing protocols of WMANs (e.g., AODV and DSR) can be integrated with the proposed RSDA protocol to achieve reliability in routing.

Mukhedkar and Kolekar [39] addressed the security issue in MANETs and proposed a secure routing protocol encrypted trust-based dolphin glowworm optimization (E-TDGO). This protocol provides security to three phases (i.e., route discovery, optimal path selection, and communication through the selected route) using a trust-based optimization model and advanced encryption standard-128 (AES-128). The trust level and the distance between nodes are utilized to discover k number of paths in the first phase to identify a normal user and an attacker. An optimal secure path is then selected from the discovered k paths using DGO novel algorithm. Communication begins in the network through an optimal path from the source to destination, and security is ensured by E-TDGO protocol.

FIGURE 3: DART logical network over the physical network.

The major challenge for PKI is to have a distributed approach for authentication of the public key in WANETs. Table 2 shows the comparative analysis of the existing security approaches. After studying thoroughly all the existing approaches for PKI in WANETs [17, 23, 24, 26–30, 31, 33–36], we have proposed a chain-based authentication for our targeted scenario. Because both DHT-based routing and WANET are distributed in nature, we have proposed a distributed PKI, as described in Section 4. The existing approaches that employ DHT mechanism for routing are mostly devised to handle the mismatch problem. A mismatch problem occurs if neighbors of a node in the logical network are different from the physical neighbors of the node. Mismatch problem increases the routing overhead and end-to-end delays when routing packets in WANETs [25]. Now we describe how the existing DHT-based routings address the mismatch problem.

Virtual cord protocol (VCP) [7] connects the nodes in a logical chord based on nodes and LIDs. VCP assumes the LID range that is 0 to 1. In the VCP node, LID value 0 does not have any predecessor node; also, a node with LID 1 does not have any successor node. VCP assigns the LID to the new joining node using its physical neighbors. Consequently, it avoids the mismatch problem.

DART [40] uses a binary tree of $M+1$ level as the logical network and allocates each node an M -bit logical identifier. The leaves of the logical tree reflect the nodes' LIDs, and the subtree reflects a node with a common prefix. DART avoids the mismatch problem by assigning physical neighbors in the same subtree structure (i.e., they share a common prefix). For example, Figure 3 shows the binary of 4-bit LID space, where nodes 0000 and 0001 have three common prefix bits at level 1. DART does not avoid completely the mismatch

problem, and the result shows that the value of the path increases 30–35% as the size of the network gets larger.

M-DART [6] is an extension of DART. It establishes a tree logical structure and maintains all of its neighboring nodes to reach the destination node. M-DART is unable to evade a complete mismatch problem due to using tree data structure [25]. Figure 3 shows the address space represented as an overlay network built upon the underlying physical topology. Its tree-based structure offers the manageable procedure for address allocation without relying on flooding. M-DART suffers from a mismatch problem due to the inflexible connecting order of its tree-based logical structure.

Caleffi et al. [41] proposed an augmented tree-based routing (ATR) protocol that provides a solution to the scalability problem. In MANETs, hierarchically organized dynamic addressing approaches use a simple and manageable tree-based structure for routing and address space management. These tree-based addressing schemes embedded incomplete information that shows unsatisfactory route selection. In the proposed ATR augment, the tree structure uses storing additional information in the node routing table that allows one to resort to multipath routing. Each node discovers all its possible paths to reach the destination using its neighbor nodes. Figure 4 shows the routing issue in path discovery results of DART and ATR with five nodes for a full mesh network. The path discovery from each node to the two destinations 2 and 4 is shown in a given graph. For the same destination graph, show the multiple paths in ATR, while existing approaches do not provide the shortest path in a very simple network. The proposed mechanism solves the medium instability or bottleneck problem and scalability issue and gains good resilience against node mobility or failure in MANETs.

FIGURE 4: Path discovery process in ATR and DART.

FIGURE 5: Hypercube $d = 4$.

Caleffi [42] present a scalable DHT-based routing protocol that integrates direct routing and indirect key-based routing at the network layer. The proposed mechanism is able to build an overlay network that improves the p2p performance by agreeing on physical and logical proximity. In addition, the presence of a hostile channel and modest node mobility assure the satisfactory performance whenever the number of nodes grows. The proposed protocol indirect tree-based routing (ITR) allocates an l bit string as a location-based identifier to the peer node. Routing is simplified in MANET; each routing table consists of an l section and a k section. Section l shows each bit of the ID, and the k section represents the address prefix length that is shared by the forwarder and destination IDs. ITR is capable to forward resource queries without introducing any overlay because logical proximity agrees with physical proximity.

Alvarez-Hamelin et al. [43] connect the nodes in logical d -hypercube structure; d refers to the dimension of a hypercube. A node has LID and mask value indicating the LID space portion for which the node is responsible. The node connected to those nodes having LID coordinates differs in one dimension. Figure 5 illustrates the connectivity of LID 0000 with the LIDs 0010, 1000, 0001, and 0100. This proto-

col still does not avoid the mismatch problem. For example, node (0110) and node 1111 are physical neighbors; they are not logical neighbors in the hypercube because their LIDs differ in more than one bit, as shown in Figure 5.

Motion mix [8] maintains 1-hop logical neighbors in an overlay network and is basically designed to handle the overhead in case of the mobility of nodes. It uses past mobility traces of nodes to predict node movement. The motion mix is partially effective against the mismatch problem.

Mesh-DHT [44] uses a 2-dimensional (2D) structure to decrease mismatching between logical and physical networks. This technique uses a link graph to build the 2D structure that is based on the methodology of [25]. The 2D closer coordinate has physically nearby neighbors that attract each other, whereas 2D distant coordinates have physically detached nodes that repulse each other. A new joining node comes to be 1-hop neighbors to the coordinate. Nodes periodically inform each other about the coordinate of its 1-hop neighbors and improve its 2D coordinates up to 2 hops away. Mesh-DHT is unable to provide an ample solution to the ill-matching problem.

Al-Mayouf et al. [45] addressed the following issues: (i) end-to-end route selection for the optimal utilization of

network resources in a VANET environment; (ii) maintaining a stable network without congestion, in the existing segment aware-based geographic routing protocol that may result in packet loss, delays, and increased communication overhead in route recovery. The authors have considered both traffic and segment status to propose a real-time interaction-based segment aware routing (RTISAR) algorithm for geographic routing in VANETs towards finding an optimal route to the destination. RTISAR considers traffic segment status when selecting the next intersection. It is based on their connectivity, density, and load of segments. The proposed scheme also considers the cumulative distance to a specific destination that can avoid selecting intersections with low connectivity, sparse density segment, high load segment, and low cumulative distance to the destination. RTISAR outperforms in terms of packet delivery ratio, packet delivery delay, and communication overhead.

Teng et al. [46] proposed a vehicles joint UAV topology discovery (VUTD) scheme to discover the physical topology with low cost and accuracy. Location information is a challenging issue for many IoT applications because most sensor devices are randomly deployed and locations are unknown. A mobile vehicle acts as a mobile anchor to assist adjacent sensor devices in positioning and also collect logical topology information of the IoT systems. Physical topology information is a combination of collected location information and logical topology information. Cloud platform receives this information via vehicles and analyzes it to determine where the physical topology discovery is incomplete. The UAVs act as a flying anchor and require the UAV fly subarea determination (UFSD) algorithm to locate/determine these points. Flight path planning algorithm based on simulating annealing (PPSA) generates a random flight path according to the neighborhood function and compares the flight distance with the old path to determine a better path for UAV flight between areas. The experimental results show that the VUTD scheme has better performance.

Li et al. [47] propose a “machine learning-based code dissemination scheme by selecting reliability mobile vehicles in 5G Networks (MLCD).” Vehicles are hard to manage in 5G networks; improving the program code coverage and its safety is a key challenge. Code disseminators will suffer a large cost of the ground control station (GCS). Therefore, the MLCD scheme chooses vehicles with high degree of reliability and coverage ratio as the code disseminator to deliver code with low cost and high accuracy rate via a genetic algorithm (GA) in a machine learning scheme. Firstly, a historical trajectory dataset is used to calculate the vehicle reliability and is selected to improve the safety degree of code dissemination. Secondly, the vehicle with a higher coverage ratio is preferred to optimize the performance of code dissemination with limited cost. Thirdly, the MLCD scheme is evaluated by both theoretical analysis and experimental studies that show improvement in safety degree of code dissemination process and coverage ratio.

Mahdi et al. [48] presented a comparative review on the clustering techniques working for efficient data aggregation in target tracking applications. Wireless sensor applications are vulnerable to energy limitation during communication.

For reducing energy consumption, there are two strategies (clustering and data aggregation) that are widely used to increase the lifetime of the sensor network. Redundant data is produced regularly in target tracking applications. To eliminate the data redundancy, there is need to deploy an effective data aggregation scheme. Authors conduct a comparative study on four existing clustering approaches including dynamic clustering, static clustering, combined clustering, and adaptive clustering. The pros and cons of these techniques are discussed for better choice that depends on various environments. The selection of an appropriate algorithm may reflect positive results in the data aggregation process.

The authors in [4, 5] propose a 3D routing protocol (3D-RP) that addresses the mismatch problem. 3D-LS provides the visualization of a newly joining node, and its associated neighboring nodes represented the main idea of 3D-RP. The 3D logical space is divided into three planes consisting of 6 dimensions and 8 octants. Each node is the resident of 3D-logical space; each node calculates the LID that imitates its intraneighbor connectivity accompanied by neighboring nodes. In 3D-RP, a node calculates its m -bit LID from 3D-LS. 3D-RP relies on local information that is obtained through *hello* messages. A node keeps a dimension parameter ($\dim$) along with its LID. The nodes are put in a group according to their dimensions that assist the packet forwarding. The following paragraph details the node joining process in 3D-RP. Figure 6 shows the Case 1 in which the newly joining node p has only one neighboring node i ; node p avails the first unoccupied dimension of node i and computes the LID based on node i 's LID. In the same way, the nodes s and q use the next two available dimensions of node i to compute their LIDs using Equation (22). Nodes s , q , and p are not physically connected in the network, so the node i 's local 3D-LS provides three different dimensions to these nodes to compute their LIDs. Similarly, the nodes l and r use the interpolation method (Equation (23)) to calculate their LIDs as shown in Case 2 of Figure 6. Also, node v uses Equation (24) to calculate its LID corresponding to s and l , which are nonadjacent neighbors as illustrated in Case 3 of Figure 6. In Case 4, the joining node t computes its LID after checking the contiguity of its neighbor nodes q , r , and i as shown in Figure 6.

Figure 7 demonstrates the intraneighbor relationship of node i with its physical neighboring nodes in the local 3D logical space of node i . The physical links are represented by black dashed lines while the 3D logical space of node i is represented by the blue dashed lines. Each node is the resident of 3D-logical space; each node calculates the LID that imitates its intraneighbor connectivity accompanied by neighboring nodes that assist in assigning nodes' LID such that the nodes that are close in the physical network obtain closest LIDs. In this way, the LID of node i is physically close to all of its adjacent neighboring nodes that shows the physical proximity in terms of x , y , and z coordinates. The exact interpretation of the physical proximity in the logical network assists in avoiding the long routes and traffic overhead redundancy and seems to reduce end-to-end delay.

To send a packet in the direction of destination LID $\{x|y|z\}$ -dim, a source node exploits its L_{nbr} that has the same

Case 1: In the case of more distinct neighbors, the remaining 3 dimensions out of the six uses the following formula to assign LID to nodes.

$$LID = \left[\begin{matrix} T_1 & T_2 & (LSP_{iy} - T_1) \end{matrix} \right] \left[\begin{matrix} T_1 & T_2 & T_3 & (LSP_{iz} + T_1) \end{matrix} \right] \left[\begin{matrix} T_1 & T_2 & T_3 & (LSP_{iz} + T_1) \end{matrix} \right]$$

$$LID = \left\{ T_{ix} \quad T_{iy} \quad \left(\frac{LSP_{iy}^-}{4} \right) \quad T_{iz} \right\} \left\{ T_{ix} \quad T_{iy} \quad T_{iz} + \left(\frac{LSP_{iz}^+}{4} \right) \right\} \left\{ T_{ix} \quad T_{iy} \quad T_{iz} + \left(\frac{LSP_{iz}^-}{4} \right) \right\} \quad (22)$$

Case 2: In the case of adjacent neighbor, each joining node compute its LID through interpolation using Equation (23). Also, the joining node uses the sign dimension of the available common contact between adjacent nodes for its LID.

$$LID = \left\{ \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{k=1}^{L_{nbr}} W_{mj}} * T_{kx} \quad \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{k=1}^{L_{nbr}} W_{mj}} * T_{ky} \quad \sum_{k=1}^{L_{nbr}} \frac{W_{mk}}{\sum_{k=1}^{L_{nbr}} W_{mj}} * T_{kz} \right\} \quad (23)$$

Case 3: In the case of non-adjacent neighbor nodes, each joining node computes its LID using Equation (24) in case the neighbor nodes have any common neighbor node. Also, the joining node sets the dim value to that of the neighbor node that is closest in terms of a number of hops.

$$LID = \left\{ \sum_{k=1}^{L_{nbr}} T_{kx} \quad \sum_{k=1}^{L_{nbr}} T_{ky} \quad \sum_{k=1}^{L_{nbr}} T_{kz} \right\} \quad (24)$$

FIGURE 6: Illustration of the node joining the algorithm of 3D-RP, where m in Equation (23) is the joining node; $L_{nbr} \geq$ represents the 1-hop logical neighboring nodes of m ; W_{mi} and W_{mj} are weights given by m to its neighboring nodes; L_{nbr} is the number of logical neighbors; T_{kx} , T_{ky} , and T_{kz} are the ordered tuples in x , y , and z dimensions of neighbor's LID.

dimension value or minimum sum of difference as the destination node and forwards the packet to destination node LID $\{x|y|z\}$ -dim. If there exists no such neighboring node, the packet is forwarded towards the “*base node*.” A node that is intricate in computing the LID of the newly joining node is referred to as the “*base node*.” 3D-RP does not provide any viable solution to handle the security issues discussed in Section 2. Moreover, the aforementioned is unable to handle network partitioning that makes it vulnerable to the critical node failures, which directly affects the network end-to-end delay and throughput.

In the past literature, the security issues have been discussed for traditional/non-DHT-based routing protocols in wireless ad hoc networks, and researchers have given various possible solutions to these issues. But deploying those solutions directly at the logically maintained network by a DHT-based routing protocol is inappropriate because the addressing, lookup, and routing in a DHT-based routing protocol is different from the traditional routing protocols that mainly rely on IP addresses for routing of packets at both the control and the data plane. Addressing the security concerns in a DHT-based routing is crucial and may completely halt the functionality of a DHT-based routing

protocol that would result in a total communication disruption among nodes. To the best of our knowledge, none of the existing literature provides a secure version of the DHT-based routing protocols that works at the network layer. This paper is the first to attempt the solution of such security threats in the context of DHT-based routing protocols that would prevent service disruption, decrease the traffic overhead, and reduce the packet loss in the network.

4. 3D Secure Routing Protocol (3D-SRP)

To counter the attacks briefed in Section 2, we have devised a secure version of the 3D routing protocol (3D-RP [5]), named as 3D secure routing protocol (3D-SRP). 3D-SRP assumes that public-private key pairs are evenly distributed among the nodes. Each node in the network has a pair of public and private keys and uses the RSA algorithm [17] for encryption and decryption. In 3D-SRP, a node authenticates a public key belonging to a particular node as follows: 3D-SRP uses an identity card (IC) and a certificate (C) for the node identification. IC comprises a public and a private component of a node as shown in Table 3. The public component

FIGURE 7: Illustration of node i 's 3D logical space and its intraneighbor relationships with its neighboring nodes in the physical network.

TABLE 3: Identity card (IC) generated by the 3D-SRP system at each node.

Identity card	
Public component	Private component
Personal identity (PI)	
Public key (PbK)	
IP address (IP)	Private key (PrK)
Logical identifier (LID)	
User signature (Sig)	

at each node refers to personal identity (PI) that uniquely identifies a node. PI includes data, e.g., UID LID, PbK_n, photograph, and user signature. Secure hash algorithm (SHA-1) is used to generate user signature [49] and is encrypted using the node's private key.

$$\text{Sig}_n = \text{Encrypt}_{\text{PrK}_n}(\text{hash}(\text{nonce})). \quad (5)$$

The private component comprises the private key (PrK_n) of a node. Security information is generated when the user introduces its PI and is stored relentlessly at the node for future usage. Certificate C_{ij} of a node i comprises of a certified IC, signed by another node j that certifies it. To get the IC signature of node i , node i 's is signed with j 's PrK_j using the summary function obtained by SHA-1. No centralized

certification authority (CA) has been used to validate IC. The integrity and authentication are automatically validated at each node by using the method that is explained in Section 4.1. Any of the trusted nodes can become the CA for a given node that builds a distributed certification authority mechanism between trusted nodes.

In 3D-SRP, each node sends a *hello* message containing IP, LID, and other related information (e.g., PbK and PI). Each *hello* message has a sequence number generated by the sending node in order to avoid the replay attack. We assume a connected network topology, i.e., network partition would not occur throughout the network operation. The proposed solution is sensitive to mobility. Handling the mobility of nodes has always been a major challenge in ad hoc networks. DHT-based routing protocols for MANETs do not support high mobility, because a logical network is maintained over the physical topology in DHT-based routing in which each node computes a logical LID in addition to its universal identifier (UID), i.e., IP/MAC address. In case the neighborhood of a node P changes or P moves to another place, precomputation of its LID reflects its relative position with respect to its new neighborhood. In case of high node mobility, the frequency of the node's LID recomputation increases which would increase traffic overhead, computation overhead, packet collision, and chances of network inconsistency. This would generate more traffic to update the network status and subsequently increases the end-to-end delay in

FIGURE 8: Illustration of the node authentication/verification and the joining process in 3D-SRP. (i) The newly joining node generates its credentials after receiving hello messages from the neighboring node(s) and sends its own public components and hash value of the identity card (IC) to the neighboring node(s). The neighbor node(s) gaze physically and use the RSS method to calculate the closeness to establish a trust. It decrypts the received information and ensures the data integrity to validate the IC; after IC validation, the neighboring node share its IC along with hash value. The newly joining node checks its validity. The verification and authentication process is a chain of trust and an issuance of the certificate after validating the ICs. (ii) The joining process of 3D-SRP is according to the joining process of 3D-RP as discussed earlier.

FIGURE 9: Establishing a chain of trust and issuance of certificate after validating the ICs.

Required: when node N_p switched on and wants to join the network, it is waiting time interval T , physical neighbor list of N_p is denoted with N_{List} and RSS method is used to measure the distance to each of its neighbor at joining node N_p .

- 1: if N_p does not receive a *Hello* message and T expires **then**
- 2: Initialize IC_p as security parameters
- 3: $LID_p \leftarrow \{1|1|1\} - 0$ $\backslash\backslash$ assign a first LID of the network to itself
- 4: **else**
- 5: Initialize IC_p as security parameters
- 6: $N_p \leftarrow N_a \in N_{List} : E(PbK_a, IC_p) || PbK_p$ $\backslash\backslash N_p$ encrypt its IC with neighbor public key & send to N_a
- 7: $N_a \leftarrow N_p : E(PbK_p, IC_a) || PbK_a$ $\backslash\backslash$ after gazing neighbor node send its credential to N_p
- 8: **end if**
- 9: Trust between p and a established after successful exchange of credential information.

ALGORITHM 1: Joining operation with authentication and verification.

the network. Therefore, we consider low mobility nodes or connected topology. The following subsections explained the functionality of every component of 3D-SRP.

4.1. Joining Process in 3D-SRP. When joining a network, a node p waits for at least three intervals of T for *hello* messages from its neighboring nodes in the network. Figure 8 illustrates the authentication and verification process of the newly joining node and the LID computation afterward based on the number of logical neighboring nodes in 3D-SRP. Upon the expiration of these intervals, the following two cases arise:

- (i) If node p does not receive any *hello* message from the existing nodes, then it assumes itself as the first node in the network. The node initializes its security parameters, i.e., its public-private keys, and assigns itself LID $\{1|1|1\}-0$ from 3D LID space; every axis,

i.e., x , y , and z , has a range from 1 to $\pm 2^M$. Node p obtains a hash value via applying a consistent hash function over its IP address and finds out the hash value closest to its LID, so it becomes the AN of itself and stores its own mapping information (i.e., LID, IP, and others)

- (ii) In the case where node p collects a *hello* message from at least one of its neighboring nodes, it initializes its public and private keys and starts the authentication process with the existing neighboring nodes as follows

To initiate the authentication process, node p chooses an existing neighbor node (e.g., node a) within its transmission range. As node p receives PbK_a of node a in the *hello* message, it sends its IC_p signed by a 's PbK_a . Upon receiving the signed IC of node p , node a then validates

Case 1

Required: information related to neighbor N_i is stored in neighbor table (NT) of N_j , and distance to N_i is measured using RSS at joining node N_j .

```

1:  if  $\exists N_i \in \text{NT} : (\exists N_m \in \text{Nbr}(N_i))! = N_j$  then           \\ neighbor count =1 then check neighbor of neighbors
2:       $\text{dim}_j \leftarrow \text{NextAvailableDim}(N_m, N_i)$            \\ if a neighbor of a neighbor found calculate LID using Equation (22)
3:  else
4:       $\text{dim}_j \leftarrow \text{first}(\text{dim}_i)$            \\ calculate LID using the first available dimension using Equation (22)
5:  end if
6:   $\text{LID}_j \leftarrow \text{ComputeLID}(N_i, \text{dim}_j)$            \\ LID computation done

```

Case 2

Required: information related to neighbors N_i and N_p is stored in neighbor table (NT) of N_j , and distance to N_i and N_p is measured using RSS at joining node N_j .

```

1:  if  $\exists N_i, N_p \in \text{NT} : N_p \in \text{Nbr}(N_i)$  and  $N_i \in \text{Nbr}(N_p)$  then           \\ adjacent neighbors found
2:       $\text{Nbr}_{\text{adj}} \leftarrow \text{true}$ 
3:  if  $\exists N_c \in \text{Nbr}(N_i)$  and  $\text{Nbr}(N_p)$  then           ** $N_c$  refers to any common neighbor other than  $N_j$ 
4:       $\text{oct}_j \leftarrow \text{NextCommonOctant}(N_c, N_i, N_p)$            \\ common neighbor found calculate LID using Equation (22)
5:  else
6:      if  $\text{Collinear}(N_j, N_i, N_p) == \text{true}$  then           \\ Collinear found
7:           $\text{LID}_j \leftarrow \text{ComputeLID}(N_i, N_p)$            \\ calculate LID using Equation (23)
8:          return
9:      else
10:          $\text{LID}_j \leftarrow \text{ComputeLID}(N_i, N_p, \text{oct}_j)$            \\ calculate the LID using next available octant using Equation (22)
11:     return
12: end if
13: endif
14: endif

```

Case 3

Required: information related to neighbors N_i and N_p is stored in neighbor table (NT) of N_j , and distance to N_i and N_p is measured using RSS at joining node N_j .

```

1:  if  $\exists N_i, N_p \in \text{NT} : N_p \notin \text{Nbr}(N_i)$  and  $N_i \notin \text{Nbr}(N_p)$  then           \\ adjacent neighbors not found
2:       $\text{Nbr}_{\text{adj}} \leftarrow \text{false}$ 
3:  if  $\exists N_c \in \text{Nbr}(N_i)$  and  $\text{Nbr}(N_p)$  then           \\ common neighbor found among neighbors
4:       $\text{LID}_j \leftarrow \text{Compute}(N_i, N_p, \text{Nbr}_{\text{adj}})$            \\ calculate LID using Equation (24)
5:  else
6:      if  $\text{Collinear}(N_j, N_i, N_p) == \text{true}$  then           \\ Collinear found
7:           $\text{LID}_j \leftarrow \text{ComputeLID}(N_i, N_p)$            \\ calculate LID using Equation (23)
8:          return
9:      else           \\ calculate the LID using next available octant using Equation (22)
10:         switch to Case 1 (Compute  $\text{LID}_j$  using the available dimension of  $N_i$  or  $N_p$ )
11:     end if
12: end if
13: endif
14: endif
15: endif

```

Case 4

Required: information related to neighbors N_i , N_p , and N_q are stored in the neighbor table (NT) of N_j , and distance to N_i , N_p , and N_q is measured using RSS at joining node N_j .

```

1:   $\text{NbrCommon} \leftarrow \text{false}$ 
2:  if  $\exists N_i, N_p, N_q \in \text{NT} : N_p \in \text{Nbr}(N_i), \text{Nbr}(N_q)$  and  $N_i \in \text{Nbr}(N_p), \text{Nbr}(N_q) N_q \in \text{Nbr}(N_i), \text{Nbr}(N_p)$  then
3:       $\text{LID}_j \leftarrow \text{ComputeLID}(N_i, N_p, N_q)$            \\ all adjacent neighbors calculate LID using Equation (23)
4:      return
5:  else if  $N_i \in \text{Nbr}(N_p), \text{Nbr}(N_q)$  and  $N_p \notin \text{Nbr}(N_q)$  and  $N_q \notin \text{Nbr}(N_p)$  or
6:       $N_p \in \text{Nbr}(N_i), \text{Nbr}(N_q)$  and  $N_i \notin \text{Nbr}(N_q)$  and  $N_q \notin \text{Nbr}(N_i)$  or
7:       $N_q \in \text{Nbr}(N_p), \text{Nbr}(N_i)$  and  $N_p \notin \text{Nbr}(N_i)$  and  $N_i \notin \text{Nbr}(N_p)$  then

```



```

9:      NbrCommon ← true
10:     LIDj ← ComputeLID(Ni, Np, Nq)      \\if neighbor is adjacent to others calculate LID using Equation (23)
11:     return
12:   else
13:     LIDj ← ComputeLID(Ni, Np, Nq, NbrCommon)  \\if neighbor is not adjacent to others calculate LID using Equation (24)
14:   end if

```

ALGORITHM 2: Joining operation (LID computation) in 3D-SRP after authentication and verification.

Required: when joining node N_p , successfully calculate its LID and *anchor node* (AN) LID after authentication and verification then store its mapping information at AN, N_p routes SII message to N_s through N_q and N_r as follows:

- 1: $N_p \leftarrow N_q \in N_{List} : SII \| E(\text{PrK}_p, \text{LID}_p, h(v), \text{SHA} - I(SII)) \| \text{PbK}_p$
- 2: $N_q \leftarrow N_r \in N_{List} : SII \| E(\text{PrK}_q, E(\text{PrK}_p, \text{LID}_p, h(v), \text{SHA} - 1(SII))) \| \text{PbK}_p \| \text{PbK}_q$
- 3: $N_r \leftarrow N_s \in N_{List} : SII \| E(\text{PrK}_r, E(\text{PrK}_q, E(\text{PrK}_p, \text{LID}_p, h(v), \text{SHA} - 1(SII)))) \| \text{PbK}_p \| \text{PbK}_q \| \text{PbK}_r$

ALGORITHM 3: Joining process in 3D-SRP store mapping information at anchor node (AN).

the received data by decrypting with its PrK_a and checks the data integrity by verifying the hashed value. This is shown through the following

$$\begin{aligned}
 \text{node}_p &\leftarrow \forall \text{node}_a \in \text{Nbr}_{List}(\text{node}_p) : \text{Encrypt}_{\text{PbK}_a}(\text{IC}_p, \text{hash}(\text{IC}_p)), \\
 \text{node}_a &\leftarrow \text{node}_p : \text{Encrypt}_{\text{PbK}_p}(\text{IC}_a, \text{hash}(\text{IC}_a)).
 \end{aligned}
 \tag{6}$$

During this process, node a ascertains node p 's trust level using physical gazing to node p (as both nodes are assumed to be close physically), which depends on whether node a knows node p or not. Node a then sends its IC_a to node p . The IC_a of node a is signed by node p 's PbK_p , which is received in node p 's IC_p . After authentication and verification of a 's IC, node p establishes the trust. Figure 9 illustrates the chain of trust and issuance of the certificate after validating the ICs.

Algorithm 1 illustrates the joining process including authentication and verification. After establishing trust with node a , node p selects another existing neighboring node for establishing mutual trust in the same manner. This process continues until node p establishes trust with all of its physical neighboring nodes. In case node p is unable to establish trust with any of its 1-hop physical neighboring nodes, then those 1-hop neighboring nodes are excluded from its neighbors' list and would not be considered for data forwarding and in the LID computation phase.

After completing the neighbor authentication, node p computes its LID as per 3D-RP specification based on its trusted physical neighboring nodes. Algorithm 2 describes the LID computation process in 3D-SRP. A node p gets the information about the neighbors of node i from the received hello message and finds that it has only one neighbor that is node p . Figure 6 shows Case 1 in which node p calculates its LID in the first unoccupied dimension of node i . In the same

way, nodes s and q compute their LIDs in two different unoccupied dimensions of node i . Node i assigns its three different dimensions to nodes p , s , and q because they are not physically connected to each other. Figure 6 shows Case 2 with the interpolation method in Equation (23) that is used to calculate the LIDs of nodes r and l after checking the adjacency with their existing neighbor nodes p and i . In the same way, node v in Case 3 of Figure 6 uses Equation (24) to calculate its LID corresponding to its neighbors s and l . The node t in Case 4 calculates its LID corresponding to its neighbors r , i , and q as shown in Figure 6.

After computing its LID, node p obtains the LID of its AN by applying a hash function over its IP address; let us say this produces the hashed value $h(v)$. The $h(v)$ is computed from the same 3D space from where the LIDs of nodes are computed. A node can act as AN for multiple nodes. Each node stores its mapping information on AN in DHT-based routing because the source node requires the LID of the destination to forward a packet. Each node performs two major operations:

- (1) One operation is performed after computing the LID to store its mapping information needed to compute the LID of its primary *anchor node* (PAN)
- (2) A node acts as a primary *anchor node* and stores the mapping information to those nodes whose LID is closest to its LID

Every node uses its 1-hop logical neighbor (L_{nbr}) information to forward the query towards the same dimension parameter of its L_{nbr} with the closest position of every tuple of its LID with the least sum of difference (LSD) to the $h(v)$ -dim. Sum of difference (SD) is computed using Equation (7) of each tuple of the L_{nbr} 's LID with the corresponding tuple of $h(v)$ -dim, and then using Equation (8) select a next-hop L_{nbr} with the LSD.

$$SD_{nbr} = (|T_{nbrx} - T_{dx}|) + (|T_{nby} - T_{dy}|) + (|T_{nbrz} - T_{dz}|), \quad (7)$$

$$LSD_{nbr} = \min_{nbr \in L_{nbr}} SD_{nbr}, \quad (8)$$

where the sum of difference is SD_{nbr} ; three tuples of nbr 's LID are T_{nbrx} , T_{nby} , and T_{nbrz} ; three tuples of hashed value or the destination's LID are T_{dx} , T_{dy} , and T_{dz} ; the least sum of difference is represented with LSD_{nbr} ; and 1-hop neighbors are represented with L_{nbr} . The node simply forwards a message to its base node if the 1-hop neighbors do not exist.

Algorithm 3 illustrates the newly joining node p storing its mapping information at the *anchor node* (AN) after LID computation and its AN LID computation. The node p sends a store-index information (SII) message that has $h(v)$ as the destination LID in order to store node p 's LID and IP address (collectively called mapping information) at the node (referred as AN of node p). The node whose LID closely matches the $h(v)$ would become the AN of node p . SII comprises node p 's LID, its IP address, $h(v)$ value, and the hash of the message $hash_{SII}$ encrypted with PrK_p as shown in Equation (9).

The $hash_{SII}$ is computed to ensure/verify the message integrity. The public key PbK_p of node p is included as plain text in the SII message. In case if a node p itself is having a LID that is closest to the $h(v)$, then p would store its own mapping information. Otherwise, the SII is forwarded towards the *anchor node* (AN) based on the destination LID (i.e., $h(v)$) as follows:

$$SII_{encrypt}(node_p) = SII \parallel \text{Encrypt}_{PrK_p}(hash_{SII}) \parallel PbK_p. \quad (9)$$

Node p then forwards the SII to a neighbor among all of its 1-hop neighboring nodes that have the closest LID to that of $h(v)$. After verifying the message integrity, the receiving node q compares the destination LID (i.e., the value of $h(v)$) with LIDs of itself and 1-hop neighbor nodes. The mapping information of node p (UID, LID, etc.) is stored by node q because its hash value $h(v)$ is closest to the LID. Otherwise, node q computes the *signature* by computing the hashed value of the message and encrypts with its own private key PrK_q . The encrypted message is then forwarded to one of its 1-hop neighbors (say node r) having the closest LID to that of value $h(v)$. The process will be repeated at every node on the path until the message arrives at p 's *anchor node*, say l .

For instance, in Figure 1, node s with LID $\{1|256|1\}$ -3 applies a hash function on the IP address of node s and computes the hash value $h(s) = \{150|-256|1\}$ -1 that is its PAN value. Node s forwards the message to the PAN as follows. Node s forwards the query to node i with LID $\{1|1|1\}$ -0 that is its *base node* because 1-hop neighbors of node s are not in dimension 1; node i has three neighbors q , t , and p with dimension 1 that is the same dimension of $h(s)$.

The sum of the difference of node i is calculated using Equation (7) as follows:

$$\begin{aligned} SD_q &= \{|100 - (150)| = 50\} + \{|-100 - (-256)| = 156\} \\ &\quad + \{|100 - 1| = 99\} \longrightarrow \{50 + 156 + 99\} \longrightarrow \{305\}, \\ SD_t &= \{|100 - (150)| = 50\} + \{|100 - (-256)| = 356\} \\ &\quad + \{|100 - 1| = 99\} \longrightarrow \{50 + 356 + 99\} \longrightarrow \{505\}, \\ SD_p &= \{|256 - (150)| = 106\} + \{|1 - (-256)| = 257\} \\ &\quad + \{|1 - 1| = 0\} \longrightarrow \{106 + 257 + 0\} \longrightarrow \{363\}, \\ SD_i &= \{|1 - 150| = 149\} + \{|1 - (-256)| = 257\} \\ &\quad + \{|1 - 1| = 0\} \longrightarrow \{149 + 257 + 0\} \longrightarrow \{406\}. \end{aligned} \quad (10)$$

Node q has the minimum LSD using Equation (8), i.e., $LSD_{nbr} = \{205\}$. So, node i forwards the SII towards node q . Finally, node q forwards the query towards node l with

$$\begin{aligned} SD_l &= \{|100 - 150| = 50\} + \{|-331 - (-256)| = 75\} \\ &\quad + \{|100 - 1| = 99\} \longrightarrow \{50 + 75 + 99\} \longrightarrow \{224\}, \end{aligned} \quad (11)$$

which is closest to the hashed value $\{150|-256|1\}$ -1. Thus, node l acts as PAN for node s . Node q becomes the secondary *anchor node* (SAN) because it is a 1-hop neighbor of node l and the second closest value to the hashed value, so node l replicates the mapping information at node q , in case PAN moves or fails and the SAN becomes active.

4.2. Lookup and Routing Process in 3D-SRP. The source node obtains the destination node's LID (say node d) from its *anchor node* before transmitting a data packet. The source node (say node s) generates a hash of the lookup request message (MREQ) and encrypts with its private key which is used as a signature as shown in Equation (12). The source node computes the $h(v)$ by applying a hash function over the node d 's IP address. Node s then sends MREQ to the *anchor node* ascribed for the node d in the same manner as a SII message is forwarded.

$$\begin{aligned} \text{MREQ of Node}_{source} &= \text{MREQ} \parallel \text{Encrypt}_{PrK_{source}}(hash_{\text{MREQ}} \parallel PbK_{source}), \\ hash_{\text{MREQ}} &= hash(SII). \end{aligned} \quad (12)$$

However, when the anchor node receives the MREQ message, it sends in reply the lookup reply message (MREP) to the source node s . The anchor node builds the MREP as shown in Equation (13). MREP includes the mapping information (i.e., LID, IP address, and public key (PbK_d)) of node d . Upon receiving the LID of the node d , node s sends the data packet towards node d in an analogous way as the SII

TABLE 4: Security evaluation of our protocol.

Attacks	How our protocol refuses the attacks
Access to user data in physical devices	User/password access
Compromised physical devices	Visual identity verification (authentication phase)
Identity impersonation	Visual identity verification (authentication phase) trust policies
Phishing, active spoofing, compromised data	Hashing and authentication Use of trusted chain
Data modification, compromised data	Hash function to guarantee data integrity
Lost data because of failure and battery discharge	Authentication
Compromised node	Use of algorithm to identify compromised nodes Change of trust level, trust elimination
Denial of service/data availability	Distributed data management and storage Distributed access to data services
Loss of connectivity	Distributed security processes Authentication

FIGURE 10: High-level Petri net model for 3D-SRP.

message is forwarded. Using the trusted chain mechanism, the node s gets the legitimate LID of the destination node d and thus avoids the destination LID tempering attack, described in Section 2.

$$\begin{aligned} \text{MREP of node}_{\text{destination}} &= \text{MREP} \parallel \text{Encrypt}_{\text{PrK}_{\text{dest}}} \text{hash}_{\text{MREP}} \parallel \text{PbK}_{\text{dest}}, \\ \text{hash}_{\text{MREP}} &= \text{hash}(\text{MREP}). \end{aligned}$$

(13)

Upon receiving the MREP, node s decrypts the message. The authenticity of the message is ensured by decrypting

the hash of the message and comparing it with the initially forwarded message. The same hash value would confirm that the message is not tempered during the course.

4.3. Updating Neighborhood and Trust Chain in 3D-SRP.

Each node periodically updates the list of its 1-hop neighboring nodes using *hello* messages. If the node examines that a neighboring node cannot be trusted anymore, it revokes the certificate of trust associated with that neighboring node and excludes it from the neighbor list. Similarly, if a node, say p , receives no *hello* messages from a neighboring node in a particular time interval, p removes that neighbor from

its neighbor list. The trust among neighboring nodes in 3D-SRP is established using the trusted chain mechanism. For instance, the trust between nodes r and p can be established by using the trusted chain, i.e., $q \longleftrightarrow p$, $r \longleftrightarrow q$, and $r \longleftrightarrow p$, where $\longleftrightarrow$ is trust. By using the trusted chain, the *anchor node* of d ensures that the LID and public key of p , i.e., PbK_p , in the SII message belong to node d , and thus, node d can be trusted. This process avoids authentication of the node's LID tempering attack, which is described in Section 2.

4.4. Security Evaluation of the Proposed Protocol. This section evaluates the proposed 3D-SRP in terms of various security attacks. The new security authentication and verification algorithms can be easily added to the 3D-SRP that makes it adaptable. To evaluate and analyze the proposed 3D-SRP in the context of practical perspective, Table 4 is added that illustrates the most frequent attacks in the self-organized wireless ad hoc networks and how the proposed 3D-SRP encounters with those attacks. It can be observed that the proposed 3D-SRP provides a high level of security and thus making the network more reliable.

5. Formal Analysis of 3D-SRP

This section presents the proposed model and its analysis for better understanding using high-level petri nets (HLPN), SMT-Lib, and Z3 solver.

5.1. Introduction to HLPN. Petri nets [50] provide a framework to model a discrete event system graphically which can be verified mathematically. High-level Petri nets are used for scientific problems with complex structure, especially stochastic, time, and fuzzy Petri nets which are used for modeling, analysis, and simulation engineering in intelligent task planning, artificial intelligence, managing symbolic and numerical information, and dynamic knowledge representation. In this paper, we have presented the proposed protocol model using high-level Petri nets. HLPN is defined as a seven-tuple $n = (p, t, f, \varphi, r, l, m_0)$, where p is a set of places; t is a set of transitions, where $p \cap t = \emptyset$; f is a flow relation, where $f \subseteq (p \times t) \cup (t \times p)$; φ maps places p to data types; transition rule defined by r ; f label represented by l ; and initial marking denoted with m_0 . $N = (p, t, f)$ is a net structure. (φ, r, l) describe the static and semantic information and ensure it is the same throughout the system. HLPN shows various types of tokens for places. These tokens are cross products of two or more types. Places in HLPN have tokens of various types. The incoming flow variables are utilized to enable a certain transition. Each transition must hold a precondition to be enabled. Similarly, transitions are fired through a postcondition that uses variables from the outgoing flows.

5.2. Formal Analysis and Verification. Formal analysis and verification process scrutinize the satisfaction level of the proposed system model by formal specification and its behavior. The bounded model describes the system in terms of rules and its properties. It is used for input param-

TABLE 5: Illustration of the data types for the proposed HLPN model.

Data type	Description
LID	LID of a node
NList	A list containing LIDs of neighboring node and neighbors of the neighboring nodes
UID	IP of a node
Hvalue	Hashed value of a node
L1	Joining node's LID
L2	Primary anchor node's LID
L3	Secondary anchor node's LID
HAS	Hashed value of UID of the anchor node
M	Message to send
PrK	The private key of a node
PbK	The public key of a node
R	A nonce (random number)
IC	Identity card of a node
C	Certificate of a node
C _{nbr}	Certificate of a neighboring node

TABLE 6: Illustration of the mappings and places defined in the proposed HLPN model.

Place	Mapping
φ (CL)	P (NList $\times$ C _{nbr})
φ (node)	P (NList $\times$ LID $\times$ PrK $\times$ PbK $\times$ C $\times$ IC)
φ (LID-UID)	P (LID $\times$ UID)
φ (NIP)	P (UID)
φ (JNH)	P (Hvalue)
φ (PS-AN)	P (L1 $\times$ L2 $\times$ L3 $\times$ Hvalue $\times$ UID $\times$ PbK)
φ (LID)	P (LID)
φ (LID-SHA)	P (LID $\times$ HSA)
φ (send)	P (M $\times$ UID)

eters on whether after the finite number of steps the system will terminate.

Microsoft Research developed the satisfiability modulo theory library (SMT-Lib) [51] with a Z3 solver to verify the proposed HLPN model. SMT-Lib used the input platform and benchmarking framework to evaluate the proposed system.

The HLPN model and verification tools check the specific properties verified by the HLPN model. HLPN model and verification tools are used to analyze the proposed HLPN model of 3D-SRP. The HLPN model shows the set P for places with a circle and set T for transition with a rectangular black box in Figure 10. Table 5 illustrates data types for the proposed HLPN model, and Table 6 shows mappings and places defined in the proposed HLPN model.

The complete working of the proposed 3D-SRP is detailed in Section 4. Here, we define formulae to map on transitions. The proposed system initiated with a joining

node receives a hello message from the existing neighboring node in its transmission range; it initializes its public and private keys and starts the authentication process with the existing neighboring nodes. The formulae in Equation (14) and Equation (15) map to the aforesaid transition:

$$\begin{aligned}
 R(\text{Key_Gene}) &= \forall z_{19} \in Z_{19}, \forall z_{20} \in Z_{20}, \\
 Z_{20}[3] &:= \text{Gene_SK}_i(z_{19}[2])\Lambda, \\
 Z_{20}[4] &:= \text{Gene_PK}_i(z_{19}[2])\Lambda, \\
 Z_{20}[5] &:= \text{Gene_g}(Z_{19}[1])\Lambda, \\
 Z_{19} &= Z_{19} \cup \{z_{20}[3], z_{20}[4], z_{20}[5]\}, \\
 Z_{20} &= Z_{20} \cup \{z_{20}[3], z_{20}[4], z_{20}[5]\},
 \end{aligned} \tag{14}$$

$$\begin{aligned}
 R(\text{CL-C}) &= \forall z_1 \in Z_1, \forall z_2 \in Z_2, \forall z_3 \in Z_3, \\
 Z_3 &= Z_3 \cup \{z_2\}.
 \end{aligned} \tag{15}$$

After completing the neighboring authentication, a node computes its LID as per the joining process of 3D-SRP specification based on its trusted physical neighboring nodes. The transition LIDs are calculated via a mathematical rule that is defined as follows:

$$\begin{aligned}
 R(\text{LID}_s) &= \forall z_4 \in Z_4, \forall z_5 \in Z_5, \\
 z_4[1] &= \text{NULL} \wedge z_4[2] := \text{LID} - \text{Root}(\)\Lambda, \\
 Z_4 &= Z_4 \cup \{z_4[1], z_4[2]\}\Lambda, \\
 z_4[1] &\neq \text{NULL} \Lambda, \\
 z_4[2] &:= \text{LID} - \text{Set}(\)\Lambda, \\
 Z_4 &= Z_4 \cup \{z_4[1], z_4[2]\}\Lambda, \\
 Z_5 &= Z_5 \cup \{z_4[2]\}.
 \end{aligned} \tag{16}$$

Every node applies a hash function over the node's UID to find the AN that stores its mapping information. The rule and transition in

$$\begin{aligned}
 R(\text{SHA}) &= \forall z_6 \in Z_6, \forall z_7 \in Z_7, \\
 z_7 &:= \text{Hsh}(z_6[2])\Lambda, \\
 Z_7 &= Z_7 \cup \{z_7\}.
 \end{aligned} \tag{17}$$

The rule in Equation (18) representing the encrypted message comprising LIDs and UIDs which are placed by the transition LPA at place LID-IP is modeled for checking purposes.

$$\begin{aligned}
 R(\text{LPA}) &= \forall z_8 \in Z_8, \forall z_9 \in Z_9, \forall z_{10} \in Z_{10}, \\
 z_9[1] &:= \text{encrypt}(z_8[2], z_8[3], z_8[4])\Lambda, \\
 z_9[2], z_9[3] &:= z_{10}\Lambda, \\
 Z_9 &= Z_9 \cup \{z_9[1], z_9[2]\}.
 \end{aligned} \tag{18}$$

TABLE 7: Simulation parameters.

Parameters	Their values
Total nodes	50
Radio range	250m
Simulation area	1000* 1000 m
Data transmission	64 pps
Total time of simulation	500 sec
Node moving speed	0.5 to 2 m/s
Traffic model	Random traffic pattern
No. of flows	18
Topology connectivity	BonnMotion 2
Propagation model	Two-ray ground
Inference model	Random waypoint

The process of selecting the anchor node in 3DSRP is depicted by the transition in

$$\begin{aligned}
 R(\text{Anchor}) &= \forall z_{11} \in Z_{11}, \forall z_{12} \in Z_{12}, \forall z_{13} \in Z_{13}, \\
 z_{13}[1] &:= z_{12}[1]\Lambda, \\
 z_{13}[4] &:= z_{11}[3]\Lambda, \\
 z_{13}[5] &:= z_{12}[2]\Lambda, \\
 Z_{13} &= Z_{13} \cup \left\{ \begin{array}{l} z_{13}[1], z_{13}[2], z_{13}[3], \\ z_{13}[4], z_{13}[5] \end{array} \right\}.
 \end{aligned} \tag{19}$$

To send a data packet, each node in 3D-SRP obtains the destination node's LID from the destination's AN. Equation (20) captures the process in the subsequent rule:

$$\begin{aligned}
 R(\text{SR}) &= \forall z_{14} \in Z_{14}, \forall z_{15} \in Z_{15}, \\
 z_{15}[1] &:= z_{14} \Lambda, \\
 \text{Encrypt}(z_{15}[2], z_{15}[3], z_{15}[4]) &:= \text{hash}(z_{14})\Lambda, \\
 Z_{15} &= Z_{15} \cup \{z_{15}[1], z_{15}[2]\}.
 \end{aligned} \tag{20}$$

Finally, the source node obtained the LID-IP pair of the destination node and forwards the data packet towards the destination node's LID. The transition is depicted by the following rule:

$$\begin{aligned}
 R(\text{SPS}) &= \forall z_{16} \in Z_{16}, \forall z_{17} \in Z_{17}, \forall z_{18} \in Z_{18}, \\
 \exists z_{17}[1]: z_{16}[1] &= z_{17}[1]\Lambda z_{18}[2] = z_{17}[5]\Lambda, \\
 Z_{18} &= Z_{18} \cup \{z_{18}[1], z_{18}[3]\}.
 \end{aligned} \tag{21}$$

(a)

(b)

FIGURE 11: Continued.

FIGURE 11: Packet delivery ratio against a number of malicious nodes in the network: (a) node mobility speed: 0.5 m/s; (b) node mobility speed: 0.1 m/s; (c) node mobility speed: 1.5 m/s; (d) node mobility speed: 2 m/s.

5.3. Verification Property. The proposed system model is verified by verification property. This property ensures that the proposed system meets the proposed specification and produces the correct result. The following property is verified:

- (1) The system computes and places the node's LID as stated by the proposed specifications

- (2) The AN safely received the encrypted requests about the destination node's LID. The AN encrypts the valid LID in the reply message, i.e., in accordance with the received UID

The proposed protocol is translated by the HLPN model and verified via the Z3 solver and SMT-Lib. The

(a)

(b)

FIGURE 12: Continued.

FIGURE 12: Normalized overhead against various numbers of malicious nodes in percentage: (a) node mobility speed: 0.5 m/s; (b) node mobility speed: 0.1 m/s; (c) node mobility speed: 1.5 m/s; (d) node mobility speed: 2 m/s.

Z3 solver verifies the performance of the HLPN model as stated by specific properties and practicability.

6. Simulation and Experimental Evaluation

The proposed (3D-SRP) algorithm is simulated using NS-2 (version 2.35). We assumed standard values of IEEE 802.11 of the last two lower layers. The simulation envi-

ronment is specified as per the values in Table 7. The proposed 3D-SRP is collated with M-DART [6], VCP [7], and simple 3D-RP [4, 5]. The performance of protocols is examined under various malicious node percentages to investigate the following metrics:

- (1) *Packet delivery ratio (PDR)*: the ratio of the total number of packets received successfully at the

(a)

(b)

FIGURE 13: Continued.

(c)

(d)

FIGURE 13: % improvement of 3D-SRP over VCP, MDART, and 3D-RP in terms of packet delivery ratio and normalized overhead. The protocol name is displayed on the first three bars of each graph, and the pattern is the same for the rest in each graph.

FIGURE 14: Computation time in 3D-SRP

destination node to the total number of packets sent by the source node.

- (2) *Normalized overhead (NO)*: the ratio of the entire routing overhead divided via successfully received total packets at the destination node.

The results are the mean estimates at 95% confidence interval. Since most of the confidence intervals are smaller, they show that our simulation results precisely represent the mean/average.

The data packets successfully delivered to the destination node are called protocol competence and termed as the packet delivery ratio. We observe the packet delivery ratio to examine the impact of malicious nodes initiating the above-mentioned attacks in the network. The percentage of malicious nodes is varied to observe the behavior of the network and protocols in the presence of mobility.

Figure 11 illustrates that 3D-SRP is capable of delivering a larger number of packets even with the increased number of malicious nodes. Although the overall overhead of 3D-SRP is slightly increased because of the effective node authentication and verification mechanism, increase in the packet delivery ratio reduces the normalized overhead (NO) for 3D-SRP as compared to 3D-RP, VCP, and MDART in the presence of malicious nodes and node mobility. This shows that in spite of the increased overhead, the capability of delivering packets in 3D-SRP is not compromised which shows the effectiveness of the proposed security mechanism in handling the security issues in the presence of malicious nodes. Moreover, 3D-SRP is an extension of 3D-RP. 3D-RP's major focus is to avoid the mismatch problem. VCP and M-DART are proven to be inefficient in handling the mismatch problem [25] that is why the impact of various attacks in terms of routing overhead is severe compared to 3D-SRP. The increased routing overhead for VCP and M-DART results in decreasing the packet delivery ratio which can be seen in Figure 11.

Normalized overhead is a key metric to determine the effectiveness of the proposed protocol. Although the overall overhead is increased for 3D-SRP because of the security measures, the normalized overhead is considerably improved when compared with VCP, M-DART, and 3D-RP because of the enhancement in the packet delivery ratio. Figure 12 illus-

trates the normalized overhead (NO) of 3D-SRP, M-DART, 3D-RP, and VCP with reference to several node mobility speeds and varying number of malicious nodes. Figure 12 demonstrates a significant gain in reducing the normalized overhead (NO) of 3D-SRP when compared with VCP, M-DART, and 3D-RP, which is evidence of the improvement of the network reliability because of the proposed security mechanism of 3D-SRP which provides reliable communication of nodes in the presence of malicious nodes, thus increasing the packet delivery ratio.

Figure 13 demonstrates the percentage improvement in normalized overhead and packet delivery ratio of 3D-SRP at numerous node mobility speeds when compared with 3D-RP, VCP, and M-DART. The improvement in the normalized overhead of 3D-SRP over M-DART, 3D-RP, and VCP is between 28 and 60%, 16 and 55%, and 42 and 70%, respectively, for a various number of malicious nodes and node mobility speeds. The increase in the number of malicious nodes and node mobility speed successfully delivered packets. The variation in the percentage improvement mismatch problem. 3D-RP solves the mismatch problem and proved to be a promising protocol compared to M-DART and VCP, where M-DART is a multipath protocol that makes it better compared to VCP. Because of the effective node authentication and verification mechanism in 3D-SRP, it emerges as more promising over 3D-RP in the existence of malevolent nodes. Thus, the percentage of PDR improvement of 3D-SRP over M-DART, 3D-RP, and VCP is 20 and 48%, 17 and 24%, and 23 and 62%, respectively, at various node mobility speeds.

To get the computation overhead for encryption and decryption mechanisms in the proposed 3D-SRP, we run RSA encryption and decryption on a system with processor Intel® Core™2 Quad CPU Q8400 at 2.66 GHz, Intel Memory (RAM) 4.00 GB, 64-bit Operating System (Windows 7 professional). The computation overhead for one operation is given in Figure 14. The computation time for key generation in RSA is the highest. However, the key is generated one time that is why its overall effect is not so significant. IC encryption and decryption are used for the authentication phase by a node with its physical neighbors. For a single hop, the encryption and decryption times for the SII message are shown in Figure 14. In case the SII message passes through

TABLE 8: Time complexity of joining operation with authentication and verification.

#	Statement	Unit cost	Operation count	Total cost	Remarks
Time complexity of Algorithm 1					
1	If N_p does not receive a hello message and at least three time interval T expires then	C_{wait}	$n-1$	$(n-1)C_{\text{wait}}$	Cost of the wait interval
2	Initialize IC_p as security parameters	C_i	$n-1$	$(n-1)C_i$	Cost of initialization (identity card and key pair generation cost)
3	$LID_p \leftarrow \{1 1 1\} - 0$	C_a	1	C_a	Cost of access
4	else	C_{LID}	1	$C_a + C_{LID}$	Cost of storing
5	Initialize IC_p as security parameters	C_i	$n-1$	$(n-1)C_i$	Cost of initialization (identity card and key pair generation cost)
6	$N_p \leftarrow N_a \in N_{\text{List}} : E(\text{PbK}_a, IC_p) \parallel \text{PbK}_p$	C_E C_{CE}	$n^2 \cdot m$ k	$(n^2 \cdot m)C_E$ $(k)C_{CE}$	Cost of encryption Cost of credential exchange
7	$N_a \leftarrow N_p : E(\text{PbK}_p, IC_a)$	C_{CE} C_{CE}	$n^2 \cdot m$ k	$(n^2 \cdot m)C_E$ $(k)C_{CE}$	Cost of encryption Cost of credential exchange
8	endif				
Time complexity of Algorithm 3					
1	$N_p \leftarrow N_q \in N_{\text{List}} : \text{SII} \parallel E(\text{PrK}_p, LID_p, h(v), \text{SHA} - \text{I}(\text{SII})) \parallel \text{PbK}_p$	C_a C_E C_{cs}	$n-1$ $n^2 \cdot m$ k	$(n-1)C_a$ $(n^2 \cdot m)C_E$ $(k)C_{cs}$	Cost to access the neighbor list Cost of encryption Cost of sending credential
2	$N_q \leftarrow N_r \in N_{\text{List}} : \text{SII} \parallel E(\text{PrK}_q, E(\text{PrK}_p, LID_p, h(v), \text{SHA} - \text{I}(\text{SII}))) \parallel \text{PbK}_q$	C_a C_E C_{cs}	$n-1$ $n^2 \cdot m$ k	$(n-1)C_a$ $(n^2 \cdot m)C_E$ $(k)C_{cs}$	Cost to access the neighbor list Cost of encryption Cost of sending credential
3	$N_r \leftarrow N_s \in N_{\text{List}} : \text{SII} \parallel E(\text{PrK}_r, E(\text{PrK}_q, LID_p, h(v), \text{SHA} - \text{I}(\text{SII}))) \parallel \text{PbK}_q \parallel \text{PbK}_r$	C_a C_E C_{cs}	$n-1$ $n^2 \cdot m$ k	$(n-1)C_a$ $(n^2 \cdot m)C_E$ $(k)C_{cs}$	Cost to access the neighbor list Cost of encryption Cost of sending credential
Adding all of the components to find out total time complexity.					
$\text{Total complexity} = (n-1)C_{\text{wait}} + (n-1)C_a + C_{LID} + (n-1)C_i + (n^2 \cdot m)C_E + (k)C_{CE} + (n^2 \cdot m)C_E + (n-1)C_a + (n^2 \cdot m)C_E + (k)C_{cs} + (n-1)C_a + (n^2 \cdot m)C_E + (k)C_{cs} = nC_{\text{wait}} - C_{\text{wait}} + 2nC_i - 2C_a + C_a + C_{LID} + 2(k)C_{CE} + 3nC_a - 3C_a + 3(n^2 \cdot m)C_E + 3(k)C_{cs} + 5(n^2 \cdot m)C_E + 2(k)C_{CE} + 3nC_a - 2C_a + 3(k)C_{cs}.$					
Worst case complexity = $O(n^2 \cdot m)$.					

n number of hops, then the total computation time will be increased by a factor of n . One can notice that the 3D-SRP algorithm has extra computational overhead for encryption and decryption processes as compared to the existing nonsecure algorithm. However, the delivery ratio and the authentication services offered by 3D-SRP are significant.

Time complexity is used to determine the performance of an algorithm for different input sizes. The time complexity of the proposed 3D-SRP is calculated using its primitive operations and their associated cost. Table 8 shows the analysis of Algorithm 1 and Algorithm 3 based on primitive operation count. Algorithm 1 comprises the decision of hello message arrival, initialization of security parameters, and the confidentiality of credentials during trust establishment, whereas Algorithm 3 shows the primitive operations of securely storing mapping information at the anchor node (AN). In the proposed 3D-SRP technique, confidentiality and authentication are achieved through public key cryptography (RSA).

RSA uses a modular exponentiation process to compute the remainder value. When the base value (i.e., plaintext/ciphertext) is raised to the exponent power (i.e., public key/private key), then the computed value is divided by the modulus value (i.e., the product of prime numbers). The cryptography algorithms use modular exponentiation due to its one-way function property. This process required $O(\text{exponent})$ time to perform exponentiation, and we represent the exponent value with m . After adding all the primitive operations, the total cost in the worst case becomes $O(n^2 \cdot m)$ after ignoring constant and lower term values.

Space complexity measures the requirement of auxiliary storage to run the proposed 3D-SRP technique. Data structure size determines the utilization of space, whereas program statements use fixed memory and they do not rely on data structures. The proposed 3D-SRP technique utilizes a tabular structure (two-dimensional array) to store information. Therefore, space complexity is (n^2) .

7. Conclusions and future work

To provide an efficient, secure, and reliable communication in a DHT-based routing protocol at the network layer in WANETs, it is imperative to address the security vulnerabilities for such protocols. The paper first highlights the major security threats that could adversely affect the performance of DHT-based routings. To counter these existing security threats, this work introduces a secure DHT-based routing protocol (3D-SRP). The proposed 3D-SRP ensures secure and reliable communication between the end users and keeps intact the resource-constrained environment (i.e., processing and energy limitations of mobile devices) of WANETs. It comprises a secure verification and authentication method to check the legitimacy of the neighboring nodes that consequently improves the packet delivery ratio of the network. In other words, 3D-SRP furnishes a secure mechanism to observe legitimate and secure routing paths for lookup and routing of packets.

In the future, the optimization of the duration of hello message and energy consumption of proposed 3D-SRP can be performed. High node mobility would be a major chal-

lenge in DHT-based routing protocols, especially in the presence of a security mechanism. The mobility issue can be explored in the future along with the security of DHT-based routing protocols. Similarly, future work can be carried out to address Sybil and Repudiation attacks through a more sophisticated approach.

Data Availability

We have performed simulation, and no dataset has been used that can be shared with the journal. The simulated results are plotted, and findings are shared in the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Huang, Q. Zhou, and D. Zhang, "Integrated wireless communication system using MANET for remote pastoral areas of Tibet," *China Communications*, vol. 13, no. 4, pp. 49–57, 2016.
- [2] N. Shah, S. A. Abid, D. Qian, and W. Mehmood, "A survey of P2P content sharing in MANETs," *Computers & Electrical Engineering*, vol. 57, pp. 55–68, 2017.
- [3] G. Al-kubati, A. Al-dubai, L. Mackenzie, and D. P. Pezaros, "Stable infrastructure-based routing for intelligent transportation systems," in *2015 IEEE International Conference on Communications (ICC)*, pp. 3394–3399, London, UK, June 2015.
- [4] S. Abid, M. Othman, and N. Shah, "Exploiting 3D structure for scalable routing in MANETs," *IEEE Communications Letters*, vol. 17, no. 11, pp. 2056–2059, 2013.
- [5] S. A. Abid, M. Othman, N. Shah, M. Ali, and A. R. Khan, "3D-RP: a DHT-based routing protocol for MANETs," *The Computer Journal*, vol. 58, no. 2, pp. 258–279, 2015.
- [6] M. Caleffi and L. Paura, "M-DART: multi-path dynamic address routing," *Wireless Communications and Mobile Computing*, vol. 11, no. 3, 409 pages, 2011.
- [7] A. Awad, R. German, and F. Dressler, "Exploiting virtual coordinates for improved routing performance in sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 9, pp. 1214–1226, 2011.
- [8] S. Shin, U. Lee, F. Dressler, and H. Yoon, "Motion-MiX DHT for wireless mobile networks," *IEEE Transactions on Mobile Computing*, vol. 15, no. 12, pp. 3100–3113, 2016.
- [9] A. Tahir, S. A. Abid, and N. Shah, "Logical clusters in a DHT-paradigm for scalable routing in MANETs," *Computer Networks*, vol. 128, pp. 142–153, 2017.
- [10] S. A. Abid, M. Othman, N. Shah et al., "Merging of DHT-based logical networks in MANETs," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 12, pp. 1347–1367, 2015.
- [11] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) routing," 2003.
- [12] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century*, pp. 62–68, Lahore, Pakistan, December 2001.

- [13] I. D. Chakeres and C. E. Perkins, "Dynamic MANET on demand (DYMO) routing protocol," *Internet-Draft Version 6*, no. IETF, 2006.
- [14] N. Shah, A. Ahmad, W. Mehmood, D. Qian, and R. Wang, "An efficient and scalable routing for MANETs," *Wireless Personal Communications*, vol. 75, no. 2, pp. 987–1004, 2014.
- [15] S. Zahid, S. A. Abid, N. Shah, S. A. N. Hussnain, and W. Mehmood, "Distributed partition detection with dynamic replication management in a DHT-based MANET," *IEEE Access*, vol. 6, pp. 18731–18746, 2018.
- [16] M. Yu, M. Zhou, and W. Su, "A secure routing protocol against byzantine attacks for MANETs in adversarial environments," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 449–460, 2009.
- [17] R. Lacuesta, J. Lloret, M. Garcia, and L. Peñalver, "A secure protocol for spontaneous wireless ad hoc networks creation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 629–641, 2013.
- [18] R. Mayrhofer, F. Ortner, A. Ferscha, and M. Hechinger, "Securing passive objects in mobile ad-hoc peer-to-peer networks," *Electronic Notes in Theoretical Computer Science*, vol. 85, no. 3, pp. 105–121, 2003.
- [19] Y. Lin, A. H. M. Rad, V. W. S. Wong, and J. H. Song, "Experimental comparisons between SAODV and AODV routing protocols," in *WMuNeP '05: Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling*, pp. 113–122, October 2005.
- [20] T. Clausen and E. Baccelli, *Securing OLSR problem statement. IETF INTERNETDRAFT, draft-clausen-manet-solsr-ps-00.txt*, 2005.
- [21] M. G. Zapata, *Secure Dynamic MANET On-Demand (SDYMO) Routing Protocol*, Tech. Univ, Catalonia, 2006.
- [22] L. Abusalah, A. Khokhar, and M. Guizani, "A survey of secure mobile ad hoc routing protocols," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 78–93, 2008.
- [23] J. Chen and J. Wu, "A survey on applied cryptography in secure mobile ad hoc networks and wireless sensor networks," in *Handbook of Research on Developments and Trends in Wireless Sensor Networks*, pp. 262–289, IGI Global, 2010.
- [24] G. Liu, Z. Yan, and W. Pedrycz, "Data collection for attack detection and security measurement in mobile ad hoc networks: a survey," *Journal of Network and Computer Applications*, vol. 105, pp. 105–122, 2018.
- [25] S. A. Abid, M. Othman, and N. Shah, "A survey on DHT-based routing for large-scale mobile ad hoc networks," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–46, 2015.
- [26] F. Zhan, N. Yao, Z. Gao, and H. Yu, "Efficient key generation leveraging wireless channel reciprocity for MANETs," *Journal of Network and Computer Applications*, vol. 103, pp. 18–28, 2018.
- [27] J. Filipek and L. Hudec, "Security architecture for the mobile ad hoc networks," *Journal of Electrical Engineering*, vol. 69, pp. 198–204, 2018.
- [28] H. Xia, J. Yu, C. L. Tian, Z. K. Pan, and E. Sha, "Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *Journal of Network and Computer Applications*, vol. 62, pp. 112–127, 2016.
- [29] D. Sathiyar and B. Gomathy, "Improved security and routing path learning in MANETs using Beer–Quiche game theoretical model in cloud computing," *Cluster Computing*, vol. 22, pp. 21–31, 2019.
- [30] S. Misra, S. Goswami, G. P. Pathak, and N. Shah, "Efficient detection of public key infrastructure-based revoked keys in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 2, 162 pages, 2011.
- [31] H. Fu, S. Kawamura, M. Zhang, and L. Zhang, "Replication attack on random key pre-distribution schemes for wireless sensor networks," *Computer Communications*, vol. 31, no. 4, pp. 842–857, 2008.
- [32] M. Yildiz, *Game theory lecture notes lectures* http://web.mit.edu/14.12/www/02F_lecture1518.pdf.
- [33] B. Rajkumar and G. Narsimha, "Trust based certificate revocation for secure routing in MANET," *Procedia Computer Science*, vol. 92, pp. 431–441, 2016.
- [34] A. R. Rajeswari, K. Kulothungan, S. Ganapathy, and A. Kannan, "A trusted fuzzy based stable and secure routing algorithm for effective communication in mobile adhoc networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 5, pp. 1076–1096, 2019.
- [35] V. Brindha, T. Karthikeyan, and P. Manimegalai, "Fuzzy enhanced secure multicast routing for improving authentication in MANET," *Cluster Computing*, vol. 22, pp. 9615–9623, 2019.
- [36] G. Arulkumaran and R. K. Gnanamurthy, "Fuzzy trust approach for detecting black hole attack in mobile adhoc network," *Mobile Networks and Applications*, vol. 24, no. 2, pp. 386–393, 2019.
- [37] X. Liu, A. Liu, T. Wang et al., "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *Journal of Parallel and Distributed Computing*, vol. 135, pp. 140–155, 2020.
- [38] T. Poongodi, M. S. Khan, R. Patan, A. H. Gandomi, and B. Balusamy, "Robust defense scheme against selective drop attack in wireless ad hoc networks," *IEEE Access*, vol. 7, pp. 18409–18419, 2019.
- [39] M. M. Mukhedkar and U. Kolekar, "E-TDGO: an encrypted trust-based dolphin glowworm optimization for secure routing in mobile ad hoc network," *International Journal of Communication Systems*, vol. 33, no. 7, article e4252, 2020.
- [40] J. Eriksson, M. Faloutsos, and S. V. Krishnamurthy, "DART: dynamic address routing for scalable ad hoc and mesh networks," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 119–132, 2007.
- [41] M. Caleffi, G. Ferraiuolo, and L. Paura, "Augmented tree-based routing protocol for scalable ad hoc networks," in *2007 IEEE International Conference on Mobile Adhoc and Sensor*, Pisa, Italy, October 2007.
- [42] M. Caleffi, "Mobile ad hoc networks: the DHT paradigm," in *2009 IEEE International Conference on Pervasive Computing and Communications*, pp. 8–9, Galveston, TX, USA, March 2009.
- [43] J. I. Alvarez-hamelin, A. C. Viana, and M. D. de Amorim, "DHT-based functionalities using hypercubes," in *Ad-Hoc Networking. IFIP International Federation for Information Processing, vol 212*, K. Agha, Ed., Springer, Boston, MA, USA, 2006.
- [44] H. Wirtz, T. Heer, R. Hummen, and K. Wehrle, "Mesh-DHT: a locality-based distributed look-up structure for wireless mesh networks," in *2012 IEEE International Conference on Communications (ICC)*, pp. 653–658, Ottawa, ON, Canada, June 2012.
- [45] Y. R. B. Al-Mayouf, N. F. Abdullah, O. A. Mahdi et al., "Real-time intersection-based segment aware routing algorithm for

- urban vehicular networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2125–2141, 2018.
- [46] H. Teng, K. Ota, A. Liu, T. Wang, and S. Zhang, “Vehicles joint UAVs to acquire and analyze data for topology discovery in large-scale IoT systems,” *Peer-to-Peer Networking and Applications*, 2020.
 - [47] T. Li, M. Zhao, and K. K. L. Wong, “Machine learning based code dissemination by selection of reliability mobile vehicles in 5G networks,” *Computer Communications*, vol. 152, pp. 109–118, 2020.
 - [48] O. A. Mahdi, A. W. Abdul Wahab, M. Y. Idna Idris et al., “A comparison study on node clustering techniques used in target tracking WSNs for efficient data aggregation,” *Wireless Communications and Mobile Computing*, vol. 16, no. 16, 2676 pages, 2016.
 - [49] *FIPS 180-1 - secure hash standard, SHA-1*, National Institute of Standards and Technology, 2015, <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf>.
 - [50] T. Murata, “Petri nets: properties, analysis and applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.
 - [51] L. De Moura and N. Bjørner, “Satisfiability modulo theories: an appetizer,” in *Formal Methods: Foundations and Applications. SBMF 2009. Lecture Notes in Computer Science*, vol. 5902, M. V. M. Oliveira and J. Woodcock, Eds., pp. 23–36, Springer, Berlin, Heidelberg, 2009.