IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 November 2022, accepted 3 January 2023, date of publication 10 January 2023, date of current version 17 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3235900

RESEARCH ARTICLE

FAPMIC: Fake Packet and Selective Packet Drops Attacks Mitigation by Merkle Hash Tree in Intermittently Connected Networks

WAQAR KHALID[1,2], NAVEED AHMAD[3], (Member, IEEE), SULEMAN KHAN[4], (Member, IEEE), NAJAM U. SAQUIB[5], MUHAMMAD ARSHAD[6], AND DURI SHAHWAR[7]

[1]School of Cyber Science and Engineering, Wuhan University, Wuhan 430000, China
[2]Institute of Management Sciences, Peshawar 25000, Pakistan
[3]Department of Computer Science, Prince Sultan University, Riyadh 12435, Saudi Arabia
[4]School of Psychology and Computer Science, University of Central Lancashire, RR1 2HE Preston, U.K.
[5]Department of Computer Science, Capital University of Science & Technology Islamabad, Islamabad 4400, Pakistan
[6]School of Cyber Science and Technology, Beihang University, Beijing 100191, China
[7]Department of Computer Science, Institute of Management Sciences Peshawar, Peshawar 25000, Pakistan

Corresponding author: Waqar Khalid (khalid.ping91@gmail.com)

This work was supported by Prince Sultan University (By paying the Article Processing Charges of this article).

ABSTRACT Delay/Disruption Tolerant Networks (DTNs) are a special category of Intermittently-ConnectedNetworks (ICNs). It has features such as long-delay, frequent-disruption, asymmetrical-data-rates, and high-bundle-error-rates. DTNs have been mainly developed for planet-to-planet networks, commonly known as Inter-Planetary-Networks (IPNs). However, DTNs have shown undimmed potency in challenged communication networks, such as DakNet, ZebraNet, KioskNet and WiderNet. Due to unique characteristics (Intermittent-connectivity and long-delay) DTNs face tough/several challenges in various research areas i.e bundle-forwarding, key-distribution, privacy, bundle-fragmentation, and malicious/selfish nodes particularly. Malicious/selfish nodes launch various catastrophic attacks, this includes, fake packet attacks, selective packet drops attacks, and denial-of-service/flood attacks. These attacks inevitably consume limited resources (persistent-buffer and bandwidth) in DTNs. Fake-packet and selective-packet-drops attacks are top among the challenging attacks in ICNs. The focus of this article is on critical analyses of fake-packet and selective-packet-drops attacks. The panoramic view on misbehavior nodes mitigation algorithms are analyzed, and evaluated mathematically through several parameters for detection probability/accuracy. This article presents a novel algorithm to detects/mitigates fake-packet and selective-packet-drops attacks. The proposed algorithm uses Merkle-Hash-Tree to detects the aforementioned attacks. The proposed algorithm added root hash along with all packets, when the malicious nodes drop packets or inject fake packets, the algorithm detects malicious nodes. Moreover, trace-driven simulation results show the proposed algorithm of this article accurately (enhanced detection-accuracy, enhanced packet delivery/packet loss ratios, and reduces false-positive/false-negative rates) detects malicious nodes which launch fake-packet and selective-packet-drops attacks, unlike previously proposed algorithms which detect only one attack (fake-packet or packet-drops at a time) or detect only malicious path (do not exactly detect malicious nodes which launch attacks). Furthermore, this article mathematically analyzed various scenarios to track exactly/position of various vehicular nodes.

INDEX TERMS Intermittently connected networks, delay tolerant networks, packet drops attack, fake packet attacks, misbehaving node, selective packet drop attacks.

I. INTRODUCTION

The emergence of intelligent-devices having short-range-wireless transmission capability has motivated the development of infrastructure-less Adhoc-Networks for the last two decades. However, traditional end-to-end based routing/forwarding protocols for Adhoc-Networks are inefficient in a challenging network environment. Because, these types of Adhoc-Networks, suffer from frequent disconnection, sparse network density, scarce resources, limited device

The associate editor coordinating the review of this manuscript and approving it for publication was Dongxiao Yu.

processing capability, and high susceptibility to security attacks. Such challenged type networks are commonly known as IntermittentlyConnectedNetworks (ICNs). There are various types of ICNs, Delay/Disruption Tolerant Networks (DTNs) are one of them [1].

DTNs are infrastructure-less Adhoc-Networks, where no end-to-end path between devices exists and disruption of nodes in the network occurs frequently [2]. DTNs are viable solution for applications suffering from intermittent-connectivity, long-delays, high-packet-error-rates, and high-packet-loss-ratios [3]. DTNs are primarily developed for planet-to-planet communication [4]. However, with advancement in wireless communication technology, DTNs have shown promising potential in emerging networks as well. Such as Vehicular-Ad-hoc-Networks (VANETs) [5], Underwater-Wireless-Sensor-Networks (UWSNs) [6], and special applications such as flood scenarios, rural area communication, and earthquakes, etc in which infrastructure is demolished due to a natural disaster [7].

With frequent disconnectivity of DTNs nodes, TCP/IP and other Adhoc-Networks protocols cannot be implemented in DTNs. That is why researchers put forward a special protocol known as bundle Protocol (BP) for DTNs. BP is used to routes/forwards a bundle in the network [8]. Moreover, BP countermeasures the challenging issues such as long/variable-delay, frequent-disruption, reliability, and communication among heterogeneous networks, by using permanent memory, custodian transfer, and convergence layer. DTNs nodes use StoreCarryForward (SCF) method to forwards packets from one node to another node [9]. The BP has the built-in capability to reliably forwards bundles/packets hop-by-hop and end-to-end. Where the convergence layer in DTNs architecture translates a bundle/packet to underline specific network architecture which is different from case to case.

Due to the specific characteristics of DTNs mentioned earlier in this article, DTNs face huge number of challenges [9]. These challenges includes, scarce-network-resources [8], bundle-routing [9], privacy [10], bundle-reliability [5], key management [11], packet-synchronization [9], bundle-security [9], and misbehaving-nodes [8], [9] particularly. Even though bundle-forwarding protocols in DTNs have been investigated adequately [12]. However, inadequate attention is paid to security loopholes (issues) in DTNs.

Researchers proposed Bundle Security Protocol (BSP) for basic security services. BSP header provides few basic security services such as confidentiality, integrity and authentication. However, in DTNs, nodes are vulnerable to large numbers of catastrophic security attacks. This includes, BlackHole/GrayHole [9], WormHole [13], DistributedDenialOfService [8], [9], [14], and malicious/selfish nodes attacks.

Malicious/Selfish nodes are one of the key challenging security issue in DTNs [15], [16]. Malicious/Selfish nodes introduce variously attacks such as Packet-Flooding [8], [9], [17], [18], Packet-Drooping (there are various categories of Packet-Dropping attacks. Such as, some misbehavior nodes drop all packets, or few misbehavior nodes drop selective packets, not all) [9], [19], and Fake-Packet attacks (FPA) [20], [21], [22], [23], [24] to overuse limited resources of networks. Moreover, this would lead to nodes unavailability, high PacketLossRatio (PLR), low PacketDeliveryRatio (PDR), and fake-packets in the network, which further degrade network performance (Due to resources consumption). Researchers proposed various algorithms for Adhoc-Networks to cope with misbehaving node attacks. However, the trivial mitigation/detection protocols of Adhoc-Networks such as, Vehicular-Adhoc Networks (VANETs) [25], Mobile-Adhoc-Networks (MANETs) [26], Wireless-Sensor-Networks (WSNs)/Underwater-Wireless-Sensor-Networks (UWSNs) [27], [28], Autonomous-Vehicles (AVs) [29], and TCP/IP [30] are not applicable in challenging ICNs such as DTNs, due to long/variable-delay, frequent-disruption, and frequent dis-connectivity of nodes in DTNs.

FPA and selective packet drop attacks (SPDA) are catastrophic attacks. These types of attacks exhaust limited resources and spread bogus packets in the networks [31]. Researchers proposed algorithms for them however, proposed solutions have some issues. Some of the researchers proposed detection algorithms, which blacklist the previous nodes in the communication path. However, the previous nodes may not always be malicious. In few articles, FPA detection algorithms have been proposed but it cannot detect intruder nodes. Some algorithms detect FPA only but at the same time, a node acts maliciously and launches SPDA, which remains undetected. This article discusses in details the aforementioned issues in the motivation and problem statement section of this article.

To address the aforementioned issues this article proposed Merkle-Hash-Tree based algorithm which exactly detects/mitigates malicious nodes which launch both FPA and SPDA. In this algorithm, initially, every node shares a public key with all nodes including trusted authority (TA). Before transferring packets, every node creates a root-hash of all packets and appends the root-hash along with packets. Forwarding nodes sign packets with a private key and make a specific packet format. The working of the proposed algorithm is discussed in the contribution section of this article. The proposed algorithm significantly improves false positive/false negative rates. The proposed algorithm of this article not only detects FPA but also detects misbehavior nodes that launch SPDA. The proposed algorithm also improves detection accuracy and resource consumption which further enhances PDR and PLR. Following are the primary contributions of this article;

- * Detection and mitigation of FPA and SPDA by only one algorithm (The proposed algorithm detects both attacks).
- * Black-listing of malicious nodes which launch FPA and SPDA (unlike previously proposed algorithms, which is path detection algorithms).

- * Mathematical Evaluation of proposed algorithms for detection accuracy and detection probability (Commonly all and particularly the proposed algorithm of this article).
- * Mathematical analyses provide theoretical idea to track various vehicular nodes (In future we will implement this idea to track the position of nodes in vehicular networks).
- * Cryptanalysis of previously proposed algorithms and the proposed algorithm of this article (this analyses highlight the cons of proposed algorithm, highlight clear idea for researcher to modify the algorithms in future).

The rest of the paper is organized as follows. Section II discusses related works on packet drop attacks and fake packet attacks. Section III discusses Motivation and Problem Statement. Section IV is related to the proposed algorithm FAPMIC. Section V is related to Mathematical Evaluation. Section VI discusses Simulation and results. Section VII is related to comparison, Followed by conclusions, and future works in the Section VIII.

II. REVIEW OF LITERATURE

Misbehavior nodes (Selfish and Malicious) are catastrophic for all types of ICNs specially DTNs, they exhaust network resources, such as buffer/memory, bandwidth, processing power, and energy resources (already outlined in this article). This section discusses existing/previously proposed algorithms which launch packet drops (misbehavior nodes that drop all packets and SPDA), and FPA.

Researchers in article [32] proposed "probabilistic misbehavior nodes detection scheme (PMDS)". The proposed algorithm detects misbehaving selfish nodes, which launch packet drops attacks (not SPDA) in DTNs. In PMDS there are two phases, one is called the event generation phase and other is known as the auditing phase. In the first phase (event generation), every event that is delegation, contact, and packet forwarding of all nodes are recorded. In the second phase (auditing phase), TA collects all event information (which is generated in the first phase) from all nodes in networks. TA passes these information (event information) from the proposed PMDS algorithm to verify malicious nodes. TA checks fewer/low reputation nodes frequently and higher reputation nodes infrequently. Proposed PMDS is a reliable scheme for selfish node detection, however high cost and suitability (the proposed scheme is not suitable, and difficult to deploy in DTNs) of the algorithm in DTNs are the downside of the proposed scheme. Also, this paper does not consider SPDA.

Researchers in articles [33], [34] proposed a particular node (known as an observer node) to safeguard the network for misbehavior packet drop attacks. The proposed scheme assume, networks monitoring node has all public keys (one node in the networks is dedicated for monitoring purpose). In bundles communication, If the "S" node forward a message to "R" node, "R" make a trust-token known as ForwarderTrustToken (FTT) to "S" and "S" also sends a trust-token known as ReceiverTrustToken (RTT) to "R" (actually nodes save encounter-history record). Both nodes sign the Token with their private keys. The observer node also calculates a group bias to detects social selfish nodes (a particular group of selfish nodes that only drops packets of other group members not from its group) in a network that launch packet drops attacks. Researchers put forward a very efficient detection algorithm. However, monitoring based algorithms are hard to implement in ICNs (due to intermittent connectivity). Also, this algorithm does not consider SPDA (this work/article considers a use case in which malicious nodes drop all packets).

Researchers in paper [35] proposed contact history based detection. The proposed algorithm detects and mitigates packet dropping attacks (Not have the ability to detects SPDA) in ICNs by using encounter-records. In this particular detection scheme, all nodes in the networks save their previous encounter-record. Nodes share their encounter-record with other nodes in the networks to detect and mitigate misbehavior nodes. The scheme detects an inconsistency in packets and misreporting of contact history (when attacker forge encounters history, proposed scheme detects). This work does not consider SPDA.

Researchers proposed reward based algorithms in [36] and [37]. The proposed algorithms detect misbehaving nodes through dedicated nodes known as OfflineSecurityManager (OSM) and VirtueBank (VB). In this scheme, OSM is a certificate authority, which issues/distributes certificates, while VB distributes credit/reward. In this algorithm when a node forwards a bundle/packet to an intermediate node, the sender node makes a "BaseLayer". This includes, IdentityOfNodes, ClassOfServic, AgreementPolicy, TimeToLive (TTL), TimeStamp, SecurityCertificate, SenderSignature, and NextForwarderNodeID. The intermediate node makes multiple "EndorseLayers" (Encrypted "BaseLayer" is called "EndorseLayer"). When the bundle is delivered to the destination, destination nodes collect information from layers ("BaseLayer", "EndorseLayers"), and forward it to "VB". The "VB" shares credit among those nodes which take part in the forwarding process. Although this is a very efficient scheme to tackle selfish nodes. This article has not been considered SPDA. Also, high processing costs and bandwidth consumption are the downsides of this algorithm.

Researchers in article [38] proposed a Watch-Dog based scheme to detect misbehaving nodes in DTNs. The proposed algorithm uses channel sensing methodology to detect packet doping misbehavior nodes (this article proposed an algorithm for packet drop attacks, however, the proposed algorithm does not detect SPDA). In this scheme when a relay node forwards a packet to an intermediate node, it keeps the packet/bundle in its storage. The relay node observes the communication channel for overhead. If the intermediate node forwards a packet to the destination, in this case the relay node compares the overhead of the communication channel to its buffer. According to the researchers, if the value of overhead matches (the algorithm assumes the intermediate node is benign or otherwise malicious. This particular node launch packet drops attacks). However, there is a probability that the intermediate node drops a bundle and forwards their own bundle, algorithm fails to detect the misbehavior node in this case. Also, there is possibility other nodes in networks (other than intermediate nodes) send messages, so in this case, there is overhead in the channel. The algorithm assumes that the intermediate node is benign, however, if the intermediate node is a misbehaving node (false negative). Although this is a very good scheme (difficult to deploy in wireless DTNs), however false positive/negative ratios are significantly high, which is the downside of this scheme. Moreover, this scheme keeps bundles in their storage after forwarding, which consume storage resources (DTNs have scarce resources), which is the downside of the proposed algorithm.

In the article [39] researchers proposed an iterative trust and reputation management system (“ITRM”). The scheme detects two types of misbehaving selfish nodes attacks. “BadMounting”, in this type of attack, “Rater (R)” (Rater is a particular node in networks which gives rating/reputation to node) decreases the reputation of “Service Provider” “(SP)”. The second category attack is known as “BallotStuffing”. In this category of attack, “R” increase the reputation of “SP”. On a positive note, in this particular scheme, the researchers derive an equation, which calculates nodes’ reputation and inconsistency. The proposed scheme compares the threshold with inconsistency. If an inconsistency is less than the threshold, the algorithm assumes it is benign, otherwise misbehavior. This article also does not considers SPDA. In article [40] researchers proposed an improvement of the work in [39] by categorizing “R”, based on the rating to “SP” (high, middle, low priority cluster). This gives a second opportunity for a node to prove that they are not misbehaving, unlike the existing scheme in article [39]. On a positive note, The false positive rates are significantly improved relative to “ITRM”. Complexity of this scheme is also improved (From Linear to Cluster in this scheme, in “ITRM” complexity is linear to nodes).

Researchers in articles [41], [42], [43] discussed different schemes to cope with misbehaving nodes (IncentiveAlgorithm, ReputationAlgorithm, GameAlgorithm). Researchers discussed the impact of misbehavior nodes on bundle delivery ratios and bundle loss ratios. Proposed algorithms do not consider SPDA. Researchers in article [44] proposed “watchdog” based collaborative-trust-management-system, which detects misbehaving nodes in natural disaster scenarios. This research article does not consider SPDA. Researchers in work [45] proposed distributed algorithm “GREAT” (Global reputation estimation and analysis technique). “GREAT” detects multiple attacks (packet drops, BallotStuffing, BadMounting). The complexity of reputation calculation makes it in-feasible for ICNs. Also proposed algorithm does not consider a group of selfish nodes which launch the SPDA.

Researchers of article [46] proposed a merkle-hash-tree (hashes calculation binary tree) and trust value for malicious nodes detection. The proposed scheme detects the SPDA in Opportunistic Networks (OppNets), which is a particular category of ICNs. The proposed algorithm calculates root-hash and appends with all packets (appends in the packet header). Destination node compares appended root-hash value with calculated root-hash (receiving node again calculates root-hash). If not matched, thus the algorithm decreases the trust value of receiving packet path (Identify path and decrease the trust value of all nodes in the path). In the case of multiple nodes in the path, it decreases the trust value of all nodes, which may be some benign nodes. This is a downside of the proposed algorithm. This leads to high false positive and false negative rates.

Researchers in [47] proposed a scheme to detect misbehaving nodes, which are responsible for the SPDA. In this particular scheme researchers proposed “HeaderField”. The proposed scheme mitigates misbehaving nodes by examining “HeaderField”. The “HeaderFiled” is also known as “IndicativeField”. “IndicativeField” is further subdivided into, “IdentificationField”, “FlagField”, and “OffsetField”. Researchers proposed an efficient scheme, however, the cost of algorithm, high ratios of false positive, and false negative are the main problems of this algorithm. Researchers in article [48] proposed a hybrid scheme (reputation and trust) to detects malicious path and misbehavior nodes, which launch the SPDA. This scheme detects misbehavior nodes by using merkle-hash-tree along with reputation (calculate direct trust and indirect trust value). In this scheme, destination nodes compare the number of bundles with hashes, if equal, researchers assume, a node is benign otherwise misbehavior (The SPDA attack is detected). However, how the bundles are counted and compared in this scheme are not mentioned in this article. High processing cost, lack of centralized node (like TA), and false positive/false negative rates are the downside of the proposed algorithm.

In article [49] the researchers proposed an algorithm for mitigation of misbehavior nodes that drop some packets and includes brand new bogus packets instead of them. The researchers proposed a packet “CreationTime” to mitigate malicious nodes. In this scheme, researchers proposed that the destination node always monitors the packet “CreationTime”. If the packet “CreationTime” of all bundles are the same or nearly the same, researchers assume the node is benign (in this case no FPA is launched) or otherwise malicious (FPA is detected). On a positive note, the authors proposed a very efficient detection algorithm. However, if the malicious nodes create a fake bundle with genuine time, algorithm cannot detects such type of malicious node. This algorithm can detects fake packets but cannot identify malicious nodes (the actual source of attacks). The researchers in work [21], [22], [23] proposed a merkle-hash-tree to detect misbehavior nodes that launch the FPA. It calculates the root-hash value with merkle-hash-tree, and then appends the root-hash value with all bundles. The destination node recalculates root-hash, if it matches with appended hash then the algorithm assumes no attack is detected, otherwise destination node assumes the FPA is detected. In the case of multiple nodes/multi-hop nodes in the communication path, the proposed algorithm assumes the last node in the communication path is malicious which may or may not be malicious.

III. MOTIVATION AND PROBLEM STATEMENT

DTNs are vulnerable to large number of security challenges which are already outlined in this article. Specifically, the dynamic-topology of nodes and the use of open networks (wireless networks, which is open for anyone to sends packets) to forward bundles offer straightforward possibilities/chances for misbehavior selfish/malicious nodes to various attacks. For example, in DTNs, misbehavior nodes can spread a huge number of false-information/fake-information into the networks. If the benign nodes further propagate these fake-packets/bogus-packets, this attack creates huge amounts of forged information/fake-information to the network. Due to limited resources of DTNs nodes, the fake-information lay a critical problem for the operation of challenging ICNs/DTNs. Furthermore, misbehavior intruder nodes launch attacks to waste precious resources, and increase throughput (Selfish nodes drop other node packets to save their resources and forward only has own packets. Sometimes misbehaving nodes drop selective packets, not all packets). The research problems (tackle research issues) on DTNs security are more challenging than conventional networks like VANETS, MANETS, and WSN (because of the unique security challenges, which are already mentioned in this article) [50]. Different from other Ad-hoc networks, and TCP/IP-based networks, DTNs represent a new network protocol architecture (bundle protocol), therefore introducing new unique security research issues/loopholes.

A. LOOPHOLES OF EXISTING DETECTION ALGORITHMS

The FPA and the SPDA are very dangerous attacks, because the aforementioned attacks waste very important resources of DTNs, and spread bogus packets in the networks, which are already stated in this paper. The researchers proposed some efficient algorithms for them. However, the proposed algorithms have some issues which are followed as.

In some research papers, researchers proposed algorithms that blacklist the previous node in the communication path, however, the previous node may or may not be malicious. The detection accuracy of the proposed algorithms are not cent percent accurate. The rate of false positive/false negative is significantly high in the proposed schemes. According to the analytic studies of this paper, few researchers proposed algorithms that detect fake-information-packets and cannot detect the actual source of attacks (malicious nodes). Few researchers proposed algorithms that accurately detect malicious/selfish nodes which launch the FPA. However, sometimes selfish/malicious nodes launch the SPDA instead of the FPA. This article proposed an algorithm that detects exactly the misbehavior nodes which launch the FPA and the SPDA (One algorithm which detects both attacks). The rates of false positive and false negative are significantly improved in our proposed algorithm (because our algorithm can detect exactly misbehavior nodes that launch the FPA, unlike other research papers, which only detect the fake-information, but do not detect the malicious nodes).

TABLE 1. Fake packet and selective packet drops attacks.

Paper	Detection-Methodology	Loop-Holes
[49]	Destination node arrange packets on the bases of packet generation time. If packet generation time is not same, algorithm assume it is misbehaving node, blacklist this node	If misbehaving node forwards bundles with legitimate time, in this case proposed scheme fail to detects. Propose scheme detects fake-packet not misbehaving intruder nodes
[21], [22], [23]	all nodes calculates root-hash for all bundles Destination node recalculates root-hash if match, so assume node is benign, otherwise malicious. algorithm blacklist all forwarder relay-nodes in the communication path.	Supposed if one node in the transmission path is malicious,that leads to whole path malicious, which is downside of the proposed algorithm
[46]	Proposed scheme calculates root-hash for packets. If root-hash not verified so algorithm decrease the trust-value of misbehaving node in the communication path by 0.1. If trust-value of malicious node becomes below 0.2, the algorithm declared it is malicious.	The trust-value of honest node is decrease if it belongs to malicious path (High false positive/negative). Detection of malicious node below the threshold value 0.2 (Why below 0.2 is malicious?)
[47]	Researcher add Header Which contains three fields: Identification, flag, offset Through Header field researchers find malicious nodes	Detects only Selective Packet Drops Attacks. More false positive/Negative ratios Extra header in Packets/Costly Difficult to reassemble packets

Table 1 summarized previously proposed algorithms (Detection Methodology and shortcoming of the previously proposed algorithms) which detect the FPA and the SPDA in ICNs. Few researchers proposed algorithms in which the destination nodes arrange packets based on the packet generation time, few researchers proposed detection algorithms which append root hash along with original packets. However, few research article proposed trust based methodology and packet header information based methodology to weed out malicious nodes. The proposed detection methodology along with loop holes are summarized in Table 1.

1) CRYPTANALYSIS OF THE PREVIOUSLY PROPOSED ALGORITHMS

As mentioned earlier in this article some of the shortcomings of previously proposed algorithms in Table 1. This section critically analyses existing proposed algorithms of the FPA and the SPDA detection/mitigation. Following are some of the possible attacks scenarios on the previously proposed algorithms of the FPA and the SPDA.

a: ATTACK SCENARIO 1

Considers an attack scenario on the previously proposed algorithms [21], [22], [23]. According to the assumption of researchers, the proposed algorithms detect the FPA when at least one fake packet reached their destination. However, according to the critical studies of this paper, this is not the case. For example, if one particular node in the networks says node "A" forwards five messages to the destination node "C" via node "B". If four messages including one fake packet reached their destination. The destination verifies the root-hash value with Merkle-Hash-Tree, the root-hash will not be verified, but this is not because of one fake packet but due to one missing packet (maybe drop packet). So according to the findings of this article, the assumption of existing algorithms are not correct, Merkle-Hash-Tree only detects the FPA in situations where all the forwarding packets reach their destination.

b: ATTACK SCENARIO 2

Consider a second attack scenario on existing proposed algorithms [21], [22], [23]. Researchers proposed algorithms in which the sender node creates the root-hash with Merkle-Hash-Tree, when the packets reached to the destination, the destination verifies the root-hash with embedded root-hash in the packet header. For example, we have four messages, M1, M2, M3, and M4, the sender nodes calculate hashes of all messages, H1, H2, H3, and H4 respectively. Furthermore, H1, H2, and H3, H4 are concatenated to calculate H12 and H34 respectively. Then H12 and H34 are concatenated to derive parent root-hash H1234. The destination nodes also calculate hashes with this process to verify the root-hash value.

However, in the opportunistic networks, some packets reached their destination with delay (due to the intermittent connectivity and long delay). If the destination concatenates H1 with H3 and H2 with H4 and further derives root-hash so obviously this root-hash will be different from the previously calculated the root-hash value. In this case, the proposed algorithms assume malicious nodes launch the FPA, however, this is not the case. Actually, in previously mentioned articles, researchers did not mention how to calculate the hashes. Unlike these papers, this article proposed an algorithm in which a specific packet format (Packet Sequence Number (PSN)) is followed, sender and the destination exactly follow the same procedure and the same PSN for the hashes calculation and verification.

c: ATTACK SCENARIO 3

Consider a third attack scenario on previously proposed algorithms [21], [22], [23]. If the malicious nodes launch both the FPA and the SPDA at the same time, the proposed algorithm detects only the FPA but do not detect the SPDA. Consider a network in which one node, say node "A" forwards three packets to the node "B", and two packets to the node "C" which is destined for the node "D". The node "B" drops one packet and the node "C" modified one packet (FPA). When the destination verifies the root-hash, so the root-hash will not be verified due to one packet drop and one fake packet. The proposed algorithms only detect the node "C" which modified packet (FPA) but do not detect the node "B" (the source of packet drop attacks).

d: ATTACK SCENARIO 4

Consider another attack scenario on previously proposed algorithms [21], [22], [23], [51]. If the malicious nodes drop one packet and forward all other packets to their destination (this attack is quite possible). When the destination node verifies the root-hash value so in this case the root-hash will not be verified with embedded root-hash in the packet header. The proposed algorithms assume the malicious nodes launch the FPA (modified the content of the message, which is an attack on the integrity of packets) but this is not the case.

e: ATTACK SCENARIO 5

Consider an attack scenario on article [46]. In this scheme, the researchers proposed an algorithm that calculates the root-hash (which is already mentioned in the literature review section of this article), if the root-hash is not verified so the proposed algorithm decreases the pre-calculated trust-value of all nodes in the communication path. Consider a possible novel attack scenario in which the malicious nodes launch the FPA. After launching attacks the malicious nodes safely change their path (Nodes are mobile in ICNs). Actually the malicious nodes aim is to launch the FPA and make a trick on the detection system, which decreases the trust-value of the honest nodes in the networks. In the future, all other nodes in the network do not trust those particular nodes in the same communication path. However in reality the nodes are benign in that particular communication path not malicious (a high false positive ratio).

f: ATTACK SCENARIO 6

Consider a colluding attack scenario (in which some misbehaving nodes launch attacks with collaboration) on previously proposed algorithms. When a malicious node "A" forwards a fake packet to another misbehaving node "B" in a different path. When the node "B" forwards that malicious packet to the destination, the destination node blacklist the transmission path of the node "B" (because previously proposed algorithms are path detection algorithms, which blacklist the last forwarder node). Which is not malicious (the destination decreased the trust-value of all nodes in that particular communication path).

Based on these observations/analyses this article concluded that researchers proposed some efficient algorithms to thwart malicious nodes which launch the FPA and the SPDA. However, every algorithm has its own merits and demerits. No perfect solution to these problems are proposed yet. This is still a big challenging issue in the DTNs. So it is urgent to propose an efficient algorithm to tackle this problem.

IV. FAPMIC: FAKE PACKET AND SELECTIVE PACKET DROPS ATTACKS MITIGATION BY MERKEL-HASH-TREE IN INTERMITTENTLY CONNECTED NETWORKS/PROPOSED SOLUTION

This article proposed an efficient algorithm that thwarts both FPA and SPDA in ICNs. The proposed algorithm significantly improves resource consumption, which ultimately enhances PDR, PLR, detection accuracy, and reduced false positive/false negative ratios (already mentioned in this paper).

In our proposed scheme this article makes the following assumptions:

* The forwarder node should use Merkle-Hash-Tree to calculates the root-hash and then automatically adds them with every bundle.
* Misbehaving nodes can drop some genuine/benign bundles (In the SPDA, malicious nodes do not drop all packets) and then inject new fake packet instead of them with brand new calculated hash value or modify the content of the packet to recalculate the root-hash value.
* Our proposed algorithm only considers the SPDA (There are so many other categories of packet drop attacks).
* For the SPDA one packet must be reached to the destination, and for the FPA all packets must be reached to the their destination.

A. SYSTEM MODEL

1) NETWORK MODEL

The network consists of thirty mobile nodes (10Cars, 10Pedestrian, 10Trams), communicating an adhoc fashion using Bluetooth. All nodes in the network have a unique key (identifier). This research works proposed that the network is loosely-time-synchronized (loosely-time-synchronized means any two nodes (A and B) are in the same time-slot any time). For the bundle authentication, this paper proposed an IdentityBasedCryptography (IBC). This paper proposed IBC, this is because of the light-overhead of IBC (ICNs have scarce resources, so IBC is suitable for ICNS). IBC generates (in IBC there is a key generator center) a valid private key for all nodes in the network from the node's identifier.

2) ADVERSARY MODEL/ATTACK SCENARIOS OF MALICIOUS NODES

Considera a FPA and a SPDA scenarios in Figure 1. Node "A", "B", "C", "D", "E", "F", "G", and "H" are DTNs nodes. For simplicity this article considers eight nodes, however in the reality there are more than eight nodes (could be any number). Node "A" have six messages, such as, M1, M2, M3, M4, M5, and M6. The node "A" forwards all the messages to node "F". However, there is no direct connection between node "A" and node "F". The node "A" forwards messages to "F" via two paths, through "C" and "D". Node "A" forwards messages M1, M2, M3 and M4 to the node "D", and forwards M5 and M6 to the node "C". The node "C" forwards message M5 and M6 as it is. But the other hand the node "D" drops two packets, M3 and M4, and makes a new fake packet instead of them. Actually the node "D" launches a FPA. Some times the node "D" drops some packets to launch a SPDA as well. On the other hand, the node "B" forwards four packets, P1, P2, P3 and P4 to the node "E", which is destined for the node "F". The node "E" drops packet P4 and makes a new fake packet instead of P4 and forwards to the node "G". The node "G" sends all packets to the node "F" (the node "G" does not knows about the node "E" makes a fake packet of P4). In this scenario the node "F" received a fake packet/information (P4 is fake packet).

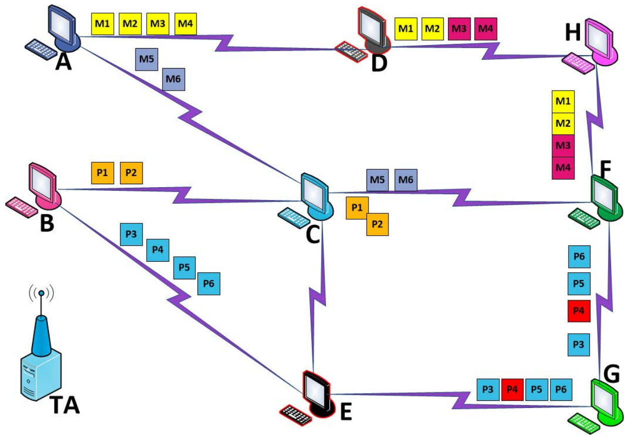


FIGURE 1. Fake Packet and Selective Packet drops attacks.

3) DEFENSE LINES AGAINST THE SPDA AND THE FPA IN FAPMIC

In this section, this article briefly states the defense mechanisms/methodology used to detects and mitigates the SPDA and the FPA. The first line of defense in our proposed algorithm is authentication/encryption. The benign nodes in the networks have their own valid cryptography-credential (Key). The nodes Sign (encryption with private key) all bundles/packets, so all other nodes in the network can authenticate the original forwarder of the bundles (a source that creates this message). Thus authentication discourages external malicious nodes (nodes that do not have a valid key) that inject unauthorized data from the outsides of the networks (outsides of the network means malicious nodes which do not have a valid key). Secondly, this authentication protects the integrity of packets (If some malicious nodes break this integrity, Our proposed scheme detects those nodes quite easily).

The second line of defense in our proposed algorithm is the creation and sending of the merkle-root-hash along with the original packets (sender side). This root-hash (created with

SHA1 Algorithm) value enables us to detect a SPDA and a FPA quite smoothly. On the receiver side, the receiving nodes compare the sender nodes' root-hash value with the calculated root-hash value (the receiver node again calculates root-hash value of all packets). The third line of defense in our proposed scheme is TA (Actually this is the action phase, this article calls this defense line because it exactly detects and blacklist the malicious nodes, which saves the network from future attacks).

B. WORKING OF PROPOSED ALGORITHM FAPMIC

This section discusses the working of the proposed algorithm of this article in detail. Before more discussion on the proposed algorithm, this article discusses the Merkle-Hash-Tree for clarity (FAPMIC uses Merkle-Hash-Tree to creates a root-hash).

1) MERKLE-HASH-TREE

In ICNs, the Merkle-Hash-Trees [21], [22] is one of the effective method to verify the integrity of the received packets. If one packet is either removed/dropped or modified/fake, the hash of its parent will change. This property of a Merkle-Hash-Tree enables us to detects a SPDA and a FPA.

A Merkle-Hash-Tree [51] is a binary-tree (Perfect binary or Complete binary both cases are possible) that starts with hashing every packet/bundle in DTNs. A Merkle-Hash-Tree uses a mathematical hash function (Hash function is one way, This article uses SHA1 for hashing) that takes a plan message/bundles and turns it into ciphertext (unique code). The resulting hashes of first-level are called the leaf of the Merkle-Hash-Tree. A pair of the leaf hashes are then concatenated (Concatenated with XOR) to derive the parent hash. A pair of parent hashes are concatenated to derive further parent hashes until the last level. The last level hash is known as root-hash which is used to check the integrity (authenticate) of all packets. In the case of a complete binary tree (a perfect binary tree, in which the number of hashes is even, and a Complete binary in which the number of hashes is odd), the odd hash is concatenated with itself to derive the parent hash. Fig 2 diagrammatically shows a Merkle-Hash-Tree. For simplicity this article considers four messages, M1, M2, M3, and M4, however in reality there are more than four packets (could be any number). In step one mathematical hash function (SHA1) are applied to M1, M2, M3, and M4 to create HASH1, HASH2, HASH3, and HASH4 respectively. Then HASH1 is concatenated to HASH2 and HASH3 is concatenated to HASH4 to create HASH5 and HASH6 respectfully. HASH5 and HASH6 are concatenated to make root-hash. Algorithm 2 is a root-hash calculation algorithm.

2) PACKET FORMAT

Figure 3 is the specific packet format in our proposed algorithm FAPMIC. A source node (Sender node, the node "A" in our case in Fig. 1) adds the source id and final destination id (Receiver of Packets, the node "F" in our case in Fig. 1). If the destination is directly connected to the source, so the source directly adds only the source id and destination id. However, if the destination is not connected directly, the node also adds in the destination column the intermediate destination id. If the node "A" forwards a packet to "D" through "C" so in this case "A" add "C" id in the intermediate destination and "D" id in the final destination field. Source and intermediate nodes encrypt the payload and the root-hash portion of the packets with their private key.

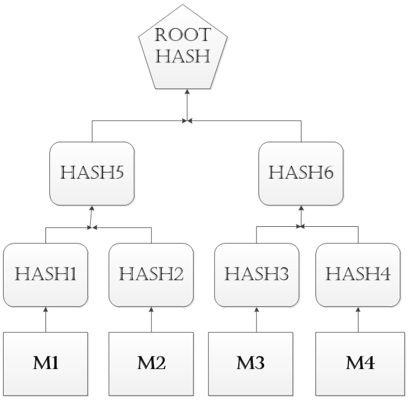
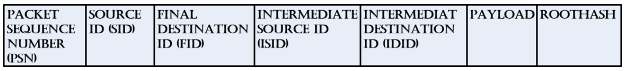


FIGURE 2. Merkle-Hash-Tree.



PACKET SEQUENCE NUMBER (PSN)	SOURCE ID (SID)	FINAL DESTINATION ID (FID)	INTERMEDIATE SOURCE ID (ISID)	INTERMEDIAT DESTINATION ID (IDID)	PAYLOAD	ROOTHASH

FIGURE 3. Packet format of fake packet and selective packet drops attack.

3) INITIAL NETWORK SETUP PHASE

During the initial network setup phase, all nodes forward their public key to TA and all other nodes in the networks. TA also shares public key with all nodes in the networks.

4) FORWARDING PHASE

In this particular phase, all nodes in the communication networks make a unique packet with a specific packet format, which is mentioned in the packet format section of this paper. Source nodes calculate root-hash with Merkle-Hash-Tree and appends with every packet. When a particular node forwards a message to another node, they keep encounter history/encounter records (Encounter is the contact between two pair devices, in DTNs contact between devices is known as encounter). Encounter history is an effective method for encounter record keeping [52]. When a node encounter other nodes, all nodes save encounter history, this includes nodes IDs (identity of both nodes), a sequence number of packets, time stamp, and sending messages history list (i.e "A" forwards packet M1 to "B" and vice-versa), both nodes sign this records with private key, (forging this records is difficult because after forging or deleting records, Encounter records becomes inconsistent, either sequence number or encounter time), for more detail refer [52]. When a destination node receives packets it saves one copy of the received message in their database (Storage).

5) ATTACKS DETECTION PHASE

When the destination node received all receiving packets. The destination decrypt messages and recalculate the root-hash value. The destination node compares the root hashes (Source hash which is included in the packet and with calculated hash), if both hashes are verified, the destination node/algorithm assumes there is no attack detected otherwise the destination node/algorithm assumes there is an attack, which is detected. If an inconsistency is found in the root-hash value, the destination sends the original copy which is saved in their database to a TA, and reports that particular misbehavior node to a TA.

A TA collects encounter-history information from all nodes. Then starts to count the number of bundles, if the number of forwarding bundles and receiving bundles are equal (TA calculates this from encounter-history, how many bundles are forwarded to that particular node, and how many bundles that particular node received). Then TA assumes there is no a SPDA detected, otherwise a SPDA is detected. If the number of received bundles is not equal to forwarding bundles so TA detects a SPDA (from encounter-history TA exactly detects malicious nodes which launch a SPDA). If TA finds that no SPDA is detected, TA starts to decrypt messages with the original key of the source, and finds all those packets which are not decrypted by the original source public key (TA has all public key). If a packet is not decrypted with the source public key, it means a malicious node drops a packet (original packet) and adds a new fake packet instead of them, TA detects a fake packet. After this TA starts to continuously decrypt that packet with all nodes' public keys sequentially. The packet will be decrypted with at least one key. The TA blacklist the signer (node) of that key is a malicious node, actually, that particular node launches a FPA. The TA exactly finds the intruder node which launches a FPA. The TA forwards blacklisting information to all nodes in the networks about misbehaving nodes. Algorithm 1 is our proposed algorithm FAPMIC.

C. CRYPTANALYSIS OF FAPMIC

1) ATTACK SCENARIO 1

Consider an attack scenario on our proposed algorithm FAPMIC. If malicious nodes launch both a FPA and a SPDA at the same time, the proposed algorithm detects only packet drops attacks (do not detect a FPA, algorithm work in this manner). For example, a node X forwards three packets to a node Y and two packets to a node Z which is destined for a node T. A node Y drops one packet (packet drops attacks) and a node Z modified one packet (FPA). When the destination verifies the root-hash, so the root-hash will not be verify due to one missing packet (packet drops) and one fake packet. The proposed algorithm only detects a node Y which drops packets (packet drops attacks) but does not detect a node Z (FPA). This is obviously the downside of our proposed algorithm (false negative, this attack is possible). In the future we will modify this algorithm for this type of attack.

Algorithm 1 FAPMIC Algorithm 1

```
0 : PhaseOne :
1 : All nodes in networks forward there public key to TA
  and all other nodes.
2 : TA shares their public key with all nodes in networks.
PhaseTwo :
3 : if have a message to forward then
    4 : Create Encounter-History packets (flush out after
      1hour) and Call algorithm 2
    Append Root-hash with Packets.
End If
5 : if Forwarding node is Final destination then
    6 : ADD Source ID and Final Destination ID.
    7 : Save one copy in Database
    8 : Destination recalculate root-hash and compare with
      packets root-hash.
    9 : if Verify root-hash then
        10 : There is no attacks detected.
        11 : Go to Step 37
    12 : else
        13 : Misbehaving nodes Attack is Detected.
        14 : Report to TA.
        15 : TA collect encounter history information from
          all nodes 16 : if number of received messages is
          not equal to forward messages then
            17 : Packet drop attack detected.
            18 : if compare all received messages SID with
              all forward messages SID. then
                19 : not malicious, Go to step 37
                20 : else
                    node is malicious,Find Missing SID in
                      receiving SID-list(Node,detection),
                    21 : Go to Step 0000 (Punishment)
                    22 : End of IF
            23 : else
                24 : FPA is detected
                25 : TA decrypt all packets with public
                  keys of original source
                26 : if decrypt then
                    27 : message is legitimate/not fake,
                      node is not malicious
                    28 : Go to 37
                    29 : else
                        Packet is Fake, node is malicious
                        30 : TA decrypt that fake packet
                          with all keys, Find a key which
                          decrypt that packet. A key which
                          decrypt packet is a node which
                          launches FPA.
                        0000: Punishment Phase: TA
                          black list the signer node
                        31 : Go to 37
        32 : else
            33 : No Packet Drop
            34 : Go to 37
    35 : else
        36 : add intermediate SID and Destination ID's and
          forward packets Go to 3
    X : End If 37 : END OF ALGORITHM
```

2) ATTACK SCENARIO 2

Consider an attack scenario on FAPMIC, a node W forwards a packet to a node K, and a node K forwards that packet to a node Y (the final destination is a node X). When a

Algorithm 2 **Root-Hash (Merkle-Hash-Tree) Calculation Algorithm 2**

```
INPUT To Algorithm: All Packets
OUTPUT of Algorithm: Root-Hash Calculation
  Through Merkle-Hash-Tree
1 : START Process to Find Hashes.
2 : Hash=HashFunction(HashP1.......HashPn) Find
  Hashes of All Packets from P1 (H1) to Pn (H2)
3 : if number Of Hashes is even (Which is already
  calculated in Step 2) then
    4 : Hash= HashFunction(HasH1 ⊕ HashH2)
      Concatenate (XOR) pair and pass to
      HashFunction
    5 : if last level then
        HashFunction(HashLastLevel ⊕
          HashLastLevel)
        ROOT-HASH=Resultant-Hash Concatenate
          Last Level to itself to derive parent Hash
        6 : End If Go to Step 9
        7 : else
            HashFuntion((HashH1 ⊕ HashH2)⊕
              HashH3)
    If odd concatenate 2 hashes and derive hash,
      derive hash is concatenated to 3rd Hash to
      further derive parent Hash Return to Step 5
    8 : End If
    9 : End Of Algorithm Return to Calling Routine in
      Algorithm 1
```

node K forwards a packet to a node Y, the packet is dropped due to some other reasons (memory overloading, Intermittent connectivity, or something else). In this case the TA blacklist a node Y (report a node Y is malicious). However, in reality, a node Y is not malicious (false positive (minor ratio), but this ratio is significantly reduced with a number of received packets, transmission range, buffer management, and processing capability).

3) ATTACK SCENARIO 3

Consider an attack scenario in which malicious nodes overloaded the buffer of the TA by flooding attacks [8], [9]. In this case, the TA will not detect attacks because our proposed algorithm FAPMIC detects attacks when the TA collects encounter history from all nodes (TA buffer is full, which cannot collects encounter history information (Single point of failure, all centralized based algorithms have this issue) this attack is quite possible (in the future we will propose an algorithm which handles this issue).

4) ATTACK SCENARIO 4

Consider a colluding attack scenario on FAPMIC. When malicious node D forwards a fake packet to another malicious node S in a different path. When a node S forwards that malicious packet to a destination, the destination node verifies the embedded root-hash value with calculated root-hash, obviously the root-hash will not be verify. However, the proposed algorithm FAPMIC does not blacklist a node S (colluding attacks are not successful on FAPMIC) because the proposed algorithm search for the actual source node which launches the attacks (FAPMIC is a node detection algorithm not a path detection algorithm. Previously proposed algorithms are path detection algorithms). This type of attacks are not successful on our proposed algorithm FAPMIC.

V. MATHEMATICAL EVALUATION

This section aims to critically analyze the proposed algorithms (Generically all and particularly our proposed algorithm FAPMIC) for detection probability and detection accuracy. This section analyzed the proposed algorithms, which ascertain some certain flaws/shortcomings of the algorithms. However, before more discussion on these observations this paper considers/assumes some assumptions, which are followed as;

All values of the constants used in the equations are based on observation/analyses (constants depend on multiple factors). The exact/accurate values of the constants and their accurate relationship with proposed parameters are beyond the scope of this paper (in the future we will make simulation-based analyses to find exact values of the constants with parameters). Table 2 shows parameters along with symbols used in the mathematical evaluation section.

TABLE 2. Parameters symbols list.

Parameter	*Symbol*	*Parameter*	*Symbol*
Detection Probability of Packet Drops Attacks	DP	Transmission Range	TRs
Detection Probability of Fake Packet Attacks	DF	Node Mobility Speed	NMS
Total Packets	TPs	Transmission Trajectory	TTs
Genuine Packets	GPs	Node Storage Space	NSS
Fake Packets	FPs	Packet Delivery Ratio	PDR
Encounter History	EHs	Speed	S
Trusted Authority	TA	Packet Size	PS
Number of Packets	NPs	Number of Nodes	NON
Detection Accuracy	DAC	Communication Area	Area
Inter Contact Time	CT	Contact Duration	CD

The detection probability and detection accuracy of the FPA and the SPDA depend on various factors (which are vary from case to case). This article analyzes some of the dependency factors (dependency factors of the detection probability and detection accuracy), which are discussed in this section.

The detection probability of the SPDA (DP) and detection probability of the FPA (DF) in our proposed algorithm are dependent on packets. The proposed algorithm detects packet drops attacks when at least one packet reach their destination out of the total packets (TPs). However, our proposed algorithm detects the FPA, when all packets are delivered to the destination. Mathematically,

$$\mathrm{DP}_{\mathrm{detected}} = 1\mathrm{Packet}/\mathrm{TPs}. \quad (1)$$

$$\mathrm{DF}_{\mathrm{detected}} = \mathrm{TPs}/\mathrm{TPs}. \quad (2)$$

While TPs are the sum of genuine packets (GPs) and fake packets (FPs).

$$\text{TPs} = \text{GPs+FPs}. \tag{3}$$

So Eq. 1 and Eq. 2 becomes,

$$\text{DP}_{\text{detected}} = 1\text{Packet}/\text{GPs+FPs}. \tag{4}$$

$$\text{DF}_{\text{detected}} = \text{GPs+FPs}/\text{GPs+FPs}. \tag{5}$$

Eq. 4 and Eq. 5 clearly show that both attacks are detected only when either one FPs or GPs reached to the destination. In case of the packet drops attacks, all GPs and FPs are delivered to the destination, however, in case of the FPA when either fake packet or genuine packet reached to their destination (the proposed algorithm FAPMIC run in this manner when either genuine or fake packet reached to the destination so algorithm starts detection process). The detection probability of our proposed algorithm also depends on encounter-history (EHs). The proposed algorithm only detects the SPDA and the FPA when TA collects EHs information from all nodes (direct relation).

$$\text{DP}_{\text{detected}} = \text{TA-Received}_{\text{EHs}}. \tag{6}$$

$$\text{DF}_{\text{detected}} = \text{TA-Received}_{\text{EHs}}. \tag{7}$$

According to the findings of this article, EHs sharing/collections depend on various factors, such as transmission-range (TRs), node-mobility-speed (NMS), transmission-trajectory (TTs), and node-storage-space (NSS), etc. According to the analyses of this article, EHs are directly proportional to TRs up to some certain limit which depend on the value of the constant.

$$\text{EHs} = \text{K1*TRs}. \tag{8}$$

In Eq. 8 K1 is a constant of proportionality, the exact value of a K1 depends on connectivity, TA/benign-nodes processing capability, and available buffer space. Eq. 8 does not imply that EHs collections are increased (unlimited increased) with TRs. In DTNs, connectivity, storage space, and processing capability have certain impact on EHs collections. TA collects EHs information from all nodes. Nodes are mobile in DTNs, when the TRs of the nodes are increased so the probability of EHs collections are also increased (there is a direct relation between TRs and encounters, which further improves the collection of EHs packets, the proof of this claim will be given in the Simulation and Results section of this paper).

According to the studies of this article, the probability of EHs sharing are directly related to NMS, because when the nodes move faster, it will increase PDR while decreasing the packet loss ratios (Because PDR is directly proportional to NMS, this will further enhance the probability of EHs sharing). For more details refer to [9].

$$\text{EHs} = \text{K2*NMS}. \tag{9}$$

When the nodes move faster, the probability of EHs are enhanced which further enhance the detection probability of the attacks. Where K2 is constant of proportionality, which depends on various factors, such as the movement direction of the nodes (when a node move with high speed, however, the direction of the movement is opposite to the TA, so in this case the probability of EHs sharing will be decreased), mobility model (MapBasedModel, RandomWayPoint, RandomWalk), and encounter (Contact with TA), etc.

According to the findings, EHs sharing with the TA also depend on NSS (Directly related). NSS is one of the most important factor for EHs sharing. Because DTNs have scarce resources (buffer space, bandwidth). If the TA buffer space is full so drops ratios become high (nodes share EHs packets, however, EHs packets are lost due to the buffer overloading) [9]. Where K4 is a constant of proportionality, which depends on the encounter, packet size (PS), number-of-packet (NPs), and node packet processing capability [8], [9].

$$\text{EHs} = \text{K4*NSS}. \tag{10}$$

while NSS is inversely proportional to PS and NPs [8], [9].

$$\text{NSS} = K5/\text{PS}. \tag{11}$$

$$\text{NSS} = K6/\text{NPs}. \tag{12}$$

And according to analyses of this article, NPs depend on the number-of-nodes (NON) [8], [9]. NPs are directly related to NON [8], [9]. Where K7 is a constant of proportionality, which depends on the node packet generation capability and node packet sending capacity.

$$\text{NPs} = \text{K7*NON}. \tag{13}$$

However, according to studies of this article, EHs sharing broadly depend on the TTs as well. EHs sharing and TTs have a direct relation with constant time K3. K3 is constant which depends on the encounter (contact with TA), the distance between nodes and TA, relative to the movement speed of TA and other nodes, TRs, node transmission speed, and direction of the packet flow, etc.

$$\text{EHs} = \text{K3*TTs}. \tag{14}$$

For illustrations/analyses this article shows various cases/scenarios of mobile nodes in the Cartesian-Plane. Fig. 4 shows various cases of the mobile nodes in Cartesian-Plane.

A. CASE1/SCENARIO1

Consider Case1 in Fig. 4, there are two nodes A, B, and TA. The TA collects EHs information from A and B. EHs collections depend on the distance between TA and nodes (which is proportional to TRs and TTs). To find the distance between node B and TA, this article considers the coordinates of the nodes in the Cartesian-Plane. The abscissa (X-Coordinate) of the TA is x2, and the ordinate (Y-Coordinate) is y2. While the X-Coordinate of B is x3 and Y-Coordinate is y3. According to the distance formula, the distance (D1, distance always positive/absolute) between TA and B will be calculated as follows.

$$\text{D1} = \sqrt{(x3 - x2)^2 + (y3 - y2)^2}. \tag{15}$$

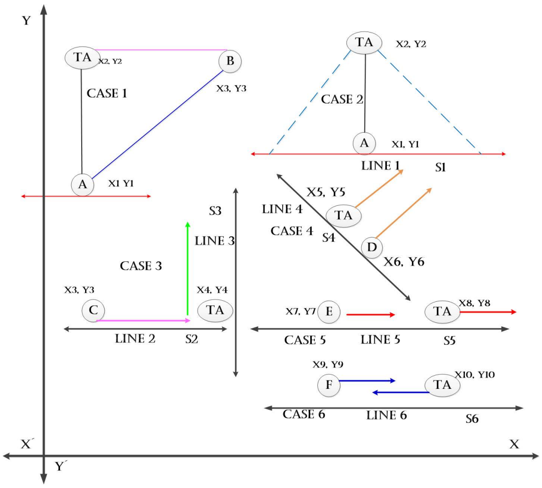


FIGURE 4. **Mathematical analysis of mobile nodes in cartesian-plane.**

The TA only collects EHs information from B with a specified distance, which depends on TRs, relative speed of TA and B. Assume TA and B move toward each other with a constant speed (S). According to the law of physics, covered distance will be calculated as follows.

$$\text{D1} = \text{S} * \text{Time (T)}. \tag{16}$$

Putting the value of Eq. 15 in Eq. 16

$$\text{S} = (\sqrt{(x3-x2)^2+(y3-y2)^2})/T. \tag{17}$$

From Eq. 17 every one can easily calculate the required moving speed of a node per unit of time.

B. CASE2/SCENARIO2

In the scenario2 of Fig. 4, the TA is located at a specific position on the Cartesian-Plane and the node is located in another specific position on the straight line (L1). The distance between the TA and the A will be calculated through a specific formula of mathematics (Distance of Point with respect to a line). Equation of the straight line is given by the following formula (according to the rules of analytic geometry, generally straight line equation).

$$\text{L1} = (ax_1 + by_1 + c). \tag{18}$$

The Coordinate of the TA are x2, y2, putting the coordinate of the TA in Eq. 18 and dividing with the $\sqrt{a^2+b^2}$, we get

$$\text{D2} = (|ax_2 + by_2 + c|)/(\sqrt{a^2+b^2}). \tag{19}$$

which is the required distance of point with respect to a line (distance between TA and node A). Putting the value of Eq. 19 in Eq. 16 we get the required speed for nodes.

$$\text{S} = (|ax_2 + by_2 + c|)/(\frac{\sqrt{a^2+b^2)}}{T}). \tag{20}$$

Putting the coordinate of the node A (x1,y1) in L1 (Eq. 18) we get the position of point (coordinate, which is actually the node position/location) with respect to a line. If the value of the line after putting the coordinate of the node A is greater than zero so the node A lies above the line, if zero, node A is on line, and if less than zero so the node A is below the line (According to the rules of mathematics).

Also, the distance between the TA and the node A can be calculated through slope (steepness). If the line between the TA and the node A (ST-line) make an anticlockwise angle with the TA (θ). So the slope of the ST-line will be calculated as follows.

$$\text{Slope (ST-line)} = \text{Tan}\theta. \tag{21}$$

$$\text{Tan}\theta = \text{Perpendicular/Base}. \tag{22}$$

Putting the value of Eq. 15 and Eq. 19 in Eq. 22 we get.

$$\text{Tan}\theta = \text{D1/D2}. \tag{23}$$

Eq. 23 easily finds the value of Tanθ, if the value of the D1 and the D2 are known. Conversely, the value of θ will be the Tan inverse of the D1 by D2 (D1/D2). Also according to the theorem of mathematics Tan θ is equal to Sinθ by Cosθ. So the Eq. 23 becomes,

$$\text{Sin}\theta = (\text{D1/D2}) * \text{Cos}\theta. \tag{24}$$

From Fig. 4 case1 Sinθ and Cosθ will be calculated as follow.

$$\text{Sin}\theta = \sqrt{(x3-x2)^2+(y3-y2)^2}/\sqrt{(y3-y1)^2+(x3-x1)^2}. \tag{25}$$

$$\text{Cos}\theta = \frac{((|ax_2+by_2+c|)}{(\sqrt{a^2+b^2}))}/(\sqrt{(y3-y1)^2+(x3-x1)^2}). \tag{26}$$

C. CASE3/SCENARIO3

In the scenario3 of Fig. 4 there are two nodes, the TA and the node C. The node C moves towards the TA, and the TA moves upwards (Perpendicular to the node C). There are various method to find the slope (steepness, which is equal to rise by run) of the given line (Slope of line is very important because the slope gives a clear idea about line, the lines are perpendicular or parallel). The coordinate of node the C and the TA are x3, y3, and x4, y4 respectively. In method one we can finds the slope of the line2 (S2) from the coordinates of the node C and the TA.

$$\text{S2} = (y4 - y3)/(x4 - x3). \tag{27}$$

In the case of line3 (L3), we know only the coordinates of the TA, so we can calculate the slope of the L3 from the line equation (general equation). Let the general equation of the L3 is,

$$\text{L3 (Y)} \Rightarrow (ax_4 + by_4 + c) = 0. \tag{28}$$

Eq. 28 can be written as (divide all terms by b),

$$\text{Y4} = (-(ax_4)/b - (c)/b). \tag{29}$$

The slope-intercept form of the general straight line equation is,

$$\text{Y} = (mx + c). \tag{30}$$

where m in Eq. 30 is the slope of the line and c is the y-intercept of the line, compare Eq. 30 and Eq. 29 to get the slope of L3 (S3).

$$S3 = (-a/b). \tag{31}$$

Multiply Eq. 27 with Eq. 31.

$$S2{*}S3 = ((-a/b) * (y4 - y3)/(x4 - x3)). \tag{32}$$

If the product of slopes of the two lines are -1 (minus one) so the lines are perpendicular or if the slopes of both lines are equal, so the lines are parallel (According to the rules of Analytic Geometry).

$$((-a/b) * (y4 - y3)/(x4 - x3)) = -1. \tag{33}$$

$$(a/b) = (1/(y4 - y3)/(x4 - x3)). \tag{34}$$

$$(-a/b) = ((y4 - y3)/(x4 - x3)). \tag{35}$$

Putting the values of the Eq. 34 in the Eq. 35 we get,

$$(-(1/(y4 - y3)/(x4 - x3))) = ((y4 - y3)/(x4 - x3)). \tag{36}$$

$$(-1) = ((y4 - y3)/(x4 - x3)) * (y4 - y3)/(x4 - x3). \tag{37}$$

D. CASE4/SCENARIO4

In the scenario4 of Fig. 4 there are two nodes, the TA and the node D. Both the nodes are moving in the same direction (parallel to each other). Relative speed and distance between the nodes are calculated from the slopes (Slope can be calculated with the same procedure mentioned earlier in this article). If the slopes of both lines are the same, this imply that the lines are parallel (According to the rules of analytic geometry). If the lines are parallel so the distance between the nodes can easily be found with a position of a point with respect to a line (Eq. 19).

E. CASE5/SCENARIO5

In the scenario5 nodes are moving in the same direction on the horizontal-axis. The slope of this line will always equal to zero (According to the rules of coordinate geometry, if the nodes move through horizontal-axis the slope will be zero (because y-coordinate is always zero, zero divided by something gives us always zero) and if the nodes move in perpendicular-axis slope will be undefined (because the x-coordinate is zero in this case. Something divided by zero gives us undefined (infinite functional value)). This is a very important result for the researchers to guess the movement direction of the nodes). For EHs collections the relative speed of the nodes are very important in this case. If the relative speed of both nodes are the same and both nodes (TA, E) are within TRs of each other, so the TA can collects EHs from the required nodes otherwise not.

F. CASE6/SCENARIO6

Consider the nodes in Fig. 4. There are two nodes, the TA and the node F, which move towards each other (Same trajectory), the slope of the nodes will be zero like the scenario5. The TA can easily collects EHs information in this case, which depend on other factors (Buffer space, processing power of the TA, etc).

The Detection Accuracy (DAC) of our proposed algorithm depends on encounters (contacts), Inter-Contact-Time (CT), Contact-Duration (CD. If nodes encounter but encounter time is very low so the probability of packet sharing is low), and NPs. DAC are inversely related to CT (nodes that meet after a long time) and directly proportional to NPs [9], [53]. NPs are directly related to NON, which is already mentioned in this article. Consider our proposed algorithm FAPMIC, the FAPMIC does not detects misbehavior nodes until the encounters of the TA and other nodes in the networks (TA collects EHs information from all nodes in the networks). If misbehavior malicious/selfish nodes meet frequently with the TA, this imply short CT, which obviously enhance the detection probability (high detection probability) otherwise the detection probability is low. However, if the number of nodes in the network is high, then the probability of the attack detection will be high (If we keep all other parameters constant).

$$\mathrm{DAC} = \mathrm{K8}/\mathrm{CT}. \tag{38}$$

$$\mathrm{DAC} = \mathrm{K9} * (\mathrm{NPs}). \tag{39}$$

$$\mathrm{DAC} = \mathrm{K10} * (\mathrm{CD}). \tag{40}$$

where K8, K9, and K10 are constants of the proportionality, depending on a mobility strategy (towards each other, moves in the opposite direction, moves parallel, or moves perpendicular to each other), mobility model, and nodes storage space. If the nodes in the networks meet frequently (encounter frequently) however, CD (duration of encounter time is not enough for a packet transmission and packet receiving) are low (low), so definitely it will significantly affect the DAC. Consider a scenario in which some nodes encounter other nodes, however, the duration of encounter is not enough (low/less) so obviously there is the possibility in which the nodes do not share an EHs bundles/packets and also the probability of the packets drops will be high in this particular case). From these observations (studies) this article concluded that the DAC is related to the communication area of the nodes (Area). The area maybe a triangle, square, circle, rectangle, and may be something else (zig-zag area, which is more probable in the communication that is why this article chose the integral area because we can find the zig-zag area very easily by the theorem of definite integral (there are two types of integral, definite and indefinite integral), we can also calculate square, circle, etc quite easily with the theorems of mathematics) According to analyses of this article the DAC is inversely related to the integral communication area.

$$\mathrm{DAC} = \frac{\mathrm{K11}}{\int_{\mathrm{P1}}^{\mathrm{P2}} (Area)^{x}}. \tag{41}$$

where P1 and P2 are two specific points in the Cartesian-Plane (any two points in the communication area). P1 and P2 are lower-bound/lower-limit and upper-bound/upper-limit of the communication area receptively. In the Eq. 41 K11 is constant. According to the studies of this article, the value of K11 depends on mobility-pattern/mobility-model, NMS, TRs, and buffer-capacity. The "x" is an integer in the above equation. The value of the "x" varies from scenario to scenario. The value of the "x" depends on the scenario (which type of scenario, how many nodes are in the scenario, and mobility model, etc), and the walking speed of the nodes in the scenario. The Eq. 41 implies that if the nodes are deployed in a small communication area (like a small cluster area, this paper simulates a scenario with ClusterMovementModel, which has a small area to prove this claim in the Simulation and Results section of this paper), this will improve the DAC and the detection probability (if the communicating nodes are deployed in a small area, so it will increase the value of the encounters, which further enhance the PDR and the detection probability. The proof of this claim will be given in the Simulation and Results section of this article).

In the Eq. 41 K11 is a mobility constant (this article calls this mobility constant), according to the findings of this paper, K11 is directly proportional to the DAC, if a node moves toward each other in the same line (opposite direction, mentioned in mathematical evaluation section) and inversely proportional, when nodes move in the same direction. We can easily solve Eq. 41 by the fundamental theorem of calculus. According to the fundamental theorem of calculus, (Area upper-limit (which is P2 in our case) - Area lower-limit (which is P1 in our case)). Let the function of the integral area is f(x) then according to the fundamental theorem of the calculus, area will be calculated as follow,

$$\mathrm{f}(x) = \mathrm{f}(P2) - f(P1). \tag{42}$$

G. MATHEMATICAL MODEL

This section aims to derive mathematical functions (relations) for the Hit-ratio and Mis-ratio of the proposed algorithm (to find the probability of the algorithm for detecting and undetected fake packets in the networks).

Let F_P be the probability that there exists one bogus packet (fake packet) in the networks, so the probability of the fake packet in the network will be calculated as follows,

$$F_P = 1 - GPs. \tag{43}$$

Let A_{hr} (Hit-ratio) be the probability of an algorithm to detects the fake packets. Let A_{mr} (Mis-ratio) be the probability of one fake packet which is undetected in one hop (surviving fake packet).

$$A_{hr} = 1 - A_{mr}. \tag{44}$$

$$A_{mr} = 1 - A_{hr}. \tag{45}$$

Let B_{mr} be the probability of one fake packet that remains undetected in all hops (AH. There are multiple hops in the networks).

$$B_{mr} = (A_{mr})^{AH}. \tag{46}$$

Let C_{mr} be the probability of N packets remaining undetected in all hops.

$$C_{mr} = (B_{mr})^{N*AH}. \tag{47}$$

The Eq. 47 clearly shows that undetected fake packets (Mis ratio) exponentially grow (up to some certain limit) with NPs and malicious hops (in which malicious nodes exist). Put the value of the Eq. 47 in the Eq. 45 we get,

$$(B_{mr})^{N*AH} = 1 - A_{hr}. \tag{48}$$

The probability of the total fake packets (TFPs) will be the sum of the detected (Hit-ratio) and undetected (Mis-ratio). From the Eq. 44 and the Eq. 45,

$$TFPs = 1 - A_{mr} + 1 - A_{hr}. \tag{49}$$

Put the value of the Eq. 48 in the Eq. 49 we get,

$$TFPs = 1 - A_{mr} + (B_{mr})^{N*AH}. \tag{50}$$

$$TFPs = A_{hr} + (B_{mr})^{N*AH}. \tag{51}$$

From the Eq. 52 anyone can find the TFPs, the detected packets, and undetected packets (it depends, if we find either hit-ratio or mis-ratio, we can easily calculate TFPs in the networks). Taking the logarithm of both sides of the Eq. 52 we gets,

$$\log(TFPs) = \log(A_{hr}) + \log(B_{mr})^{N*AH}. \tag{52}$$

$$\log(TFPs) = \log(A_{hr}) + (N*AH)*\log(B_{mr}). \tag{53}$$

$$\log(A_{hr}) = \log(TFPs) - (N*AH)*\log(B_{mr}). \tag{54}$$

VI. SIMULATION AND RESULTS

This paper has evaluated the performance of our proposed algorithm FAPMIC for the misbehaving nodes detection in ICNs through various evaluation techniques. Evaluation is done with the help of simulation. This paper simulates the proposed algorithm FAPMIC in the Opportunistic Network Environment (ONE) [54] simulator, ONE is specially designed for the ICNs (DTNs). This paper compared the proposed algorithm FAPMIC with previously proposed algorithms. Simulation is carried out on various metrics (parameters) given in the Table below. Table 3 shows the parameters-list for our proposed algorithm FAPMIC (Same simulation parameters setup with previously proposed algorithms).

A. EVALUATION-METRICS

Simulation is evaluated based on various proposed metrics rigorously, which are followed as.

1) PDR AND PLR

It is the ratio between delivered bundles to the total generated bundles. If the number of delivered bundles are DB and the total generated bundles are TGB then PDR and PLR will be measured as follow.

$$\mathrm{PDR} = (\mathrm{DB}/\mathrm{TGB}) * 100. \tag{55}$$

$$\mathrm{PLR} = ((\mathrm{TGB} - \mathrm{DB})/\mathrm{TGB}) * 100. \tag{56}$$

TABLE 3. List of simulation-parameters.

Parameter	Value	Parameter	Value
Mobility-Model	RWP,RWM,CMM	TTL	300
Area	500 * 500	WaitTime	10, 30
Packet-Size	500 to 700 K	Router	Epidemic
G1,G2,G3	Pedestrian,Car,Tram	Router	DirectDelivery
Moving-Speed (G1,G2,G3)	(1,5),(35,60),(25,35)	Router	FirstContact
Transmit-Range	10,20,30 M	Router	SprayAndWait
Simulation Time	13Hour	Groups	3
Number-of-Nodes	30	UpdateInterval	0.5
Simulator	ONE	TransmitSpeed	8M

2) LATENCY

A specific amount of time required from the creation of bundles to delivery to the destination is known as latency (this paper calculates average latency in the simulation).

3) PACKET-AVERAGE-BUFFER-TIME (PABT)

PABT is the duration of time that packets spend in the memory. PABT is an important parameter to judge the efficiency of all algorithms because PDR/PLR and detection of misbehaving attacks depend on PABT (Our proposed algorithm detects attacks when the TA collects EHs, which need space in the buffer, that is why this article considers PABT).

4) TOTAL ENCOUNTERS (TEs)

TEs are the total-numbers-of-encounters (contact/meeting) of all nodes in the simulation.

5) DETECTION ACCURACY

The Detection Accuracy is the ratio of misbehaving nodes' attack packets that are accurately detected out of all the attack packets.

6) DETECTION DELAY

The average amount of time required to detects the first malicious packet/malicious node.

7) WASTED TRANSMISSION (WT)/BANDWIDTH CONSUMPTION

It is the average amount of wasted transmission (wastage of bandwidth) in the simulation times. If the total relayed bundles/packets are TRPs, the total aborted bundles/packets are TAPs (This includes both forwarder side and receiver side aborted bundles), the total forwarder-side aborted packets TFAPs, and the total receiver-side aborted packets are TRAPs. The size of the bundles in Kb are (SPs) then the the total bandwidth consumption will be calculated as follow,

$$WT = ((TRPs) + (TAPs)) * (SPs))/1000. \tag{57}$$

In the above equation 1000 directly convert packets from kilobytes to megabytes. In the Eq. 57, TAPs includes those bundles/packets which are suddenly aborted after the relay phase. Actually, TAPs consume the transmission (bandwidth), however, this article does not consider these bundles in the packet drops. According to the analyses of this article aborted bundles are divided into two broad categories/types. Such as forwarder-side aborted and receiver-side aborted bundles. TAPs in the above equation are forwarder-side aborted packets (This article only considers this simulation case. This article considers various packet sizes, however, for demonstration purposes this article shows wasted transmission results with packet size being 20K constant (Multiply our simulation results with 35 to convert it to 700K packet size)). If TAPs are receiver side aborted then Eq. 57 will become

$$WT = ((TRPs) + ((TFAPs) - TRAPs) * (SPs))/1000. \tag{58}$$

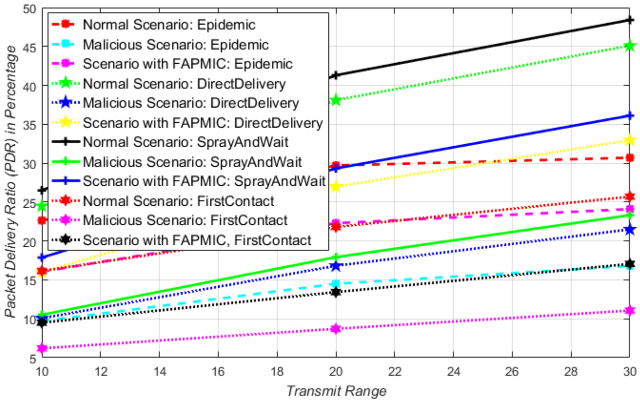


FIGURE 5. PDR of routing protocols with transmit range.

8) FALSE POSITIVE

Categorizing benign nodes as malicious nodes.

9) FALSE NEGATIVE

Categorizing malicious nodes as innocent nodes.

B. SIMULATION RESULTS

This article rigorously checked the proposed parameters by various methods/tests. Results gained for the proposed algorithm FAPMIC is discussed as follows.

1) EXPERIMENT 01

This paper in "experiment 01" evaluated/tested PDR, AL, WT, PABT, and TEs for the nodes mobility models, such as RandomWalkModel (RWM), RandomWayPointModel (RWP), and ClusterMovementModel (CMM) to evaluates our proposed algorithm FAPMIC for the SPDA.

Fig. 5, Fig. 6 and Fig. 7 demonstrate testing results of the PDR of routing protocols with RWM, RWP and CMM respectively. Just for the demonstration purposes this paper only shows testing results of two routing protocols i.e Epidemic and SparyAndWait in RWP and CMM. This article simulates routing protocols without malicious nodes (normal scenario), a scenario with malicious nodes (malicious nodes

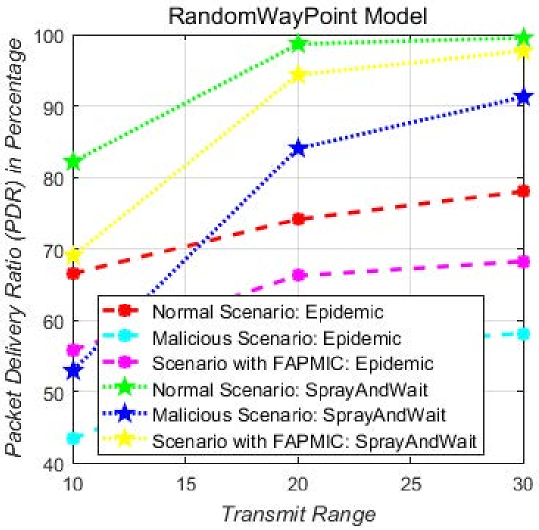


FIGURE 6. PDR With transmit range. RWP model.

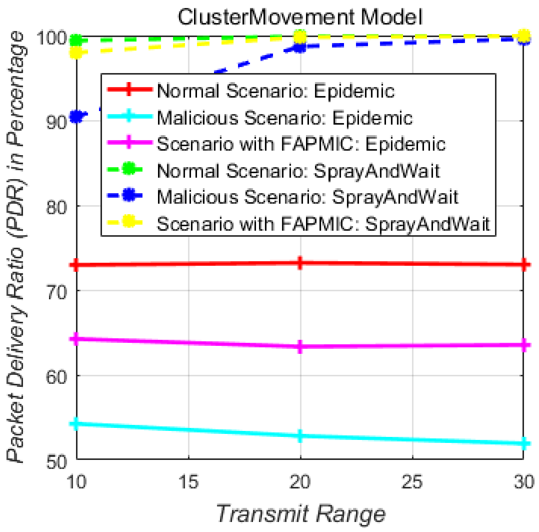


FIGURE 7. PDR with transmit range. CMM model.

which launch SPDA), and a malicious scenario with our proposed algorithm FAPMIC.

Simulation results clearly show, PDR is decreased (In all simulating routing protocols) while the PLR is increased (For demonstration purposes this paper shows testing results of PDR due to page limitation. Everyone can easily calculate PLR from these results (formula given in this paper)). This is because of malicious nodes which launch the SPDA. Experimental results clearly illustrate, the ratios of PDR is improved with our proposed algorithm FAPMIC because the FAPMIC detects and blacklists misbehavior nodes that launch SPDA. This ultimately enhanced PDR and PLR ratios.

From experimental results, this research works observed that due to the SPDA, approximately 9 to 25 percent, 9 to 29 percent, and 02 to 18 percent PDR decreases with RWM, RWP, and CMM respectively. Simulation results clearly show that approximately 3.3 to 12 percent (RWM), 6 to 16 percent (RWP), and 02 to 10 percent (CMM) PDR is improved due to our proposed algorithm. Experimental results clearly demonstrate that SprayAndWait is mostly affected while FirstContact is less affected due to the misbehavior nodes attacks. Because SprayAndWait spray packets then wait some certain times, the proposed algorithm does not detect malicious nodes in the wait-time. FirstContact forwards only packets to the FirstContacted node (sends fewer packets) and the probability of the detection is high that is why FirstContact is less affected due to the SPDA.

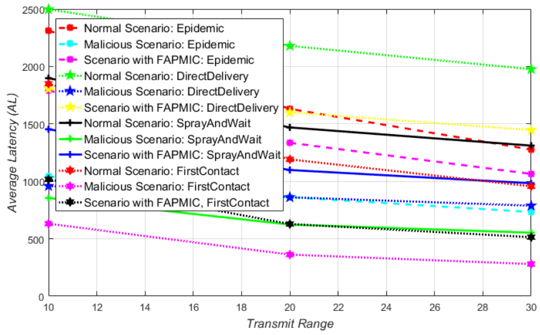


FIGURE 8. Average latency of routing protocols with transmit range.

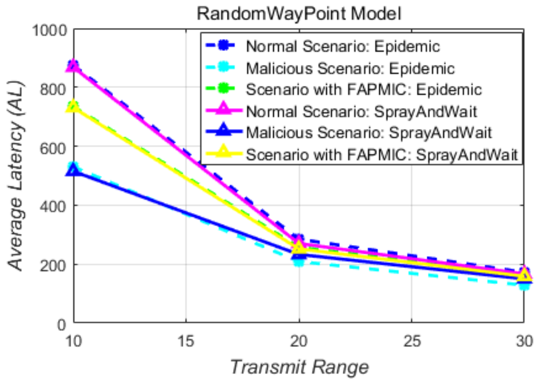


FIGURE 9. Average latency with transmit range. RWP model.

Fig. 8, Fig. 9 and Fig. 10 shows “Experiment 01” results of average latency (AL) of routing protocols with various transmit ranges. Mobility-Model are RWM, RWP, and CMM respectively in the aforementioned figures. Simulation results clearly show that AL is high when no malicious attacks, and the graphs become down when the malicious nodes launch attacks. This is because some packets are dropped, which are not delivered to the destination (More packets are created but fewer packets are delivered to the destination due to the SPDA, which is why graphs suddenly down). Simulation results clearly show that the proposed algorithm improved AL (The proposed algorithm graph is above malicious graphs it seems AL is dis-improves but in reality, this is an improvement because the proposed algorithm detects malicious nodes and enhances PDR which is why AL is high). Fig. 11, Fig. 12 and Fig. 13 shows

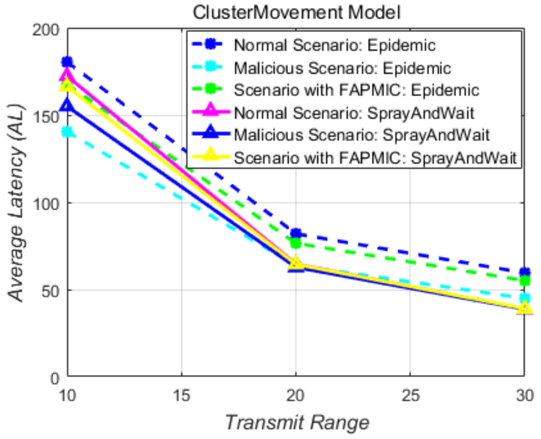


FIGURE 10. Average latency with transmit range. CMM model.

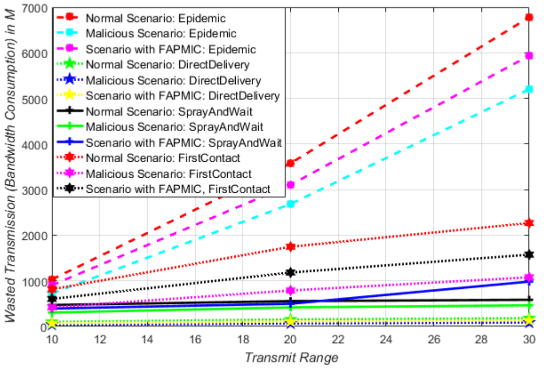


FIGURE 11. Wasted transmission of routing protocols with transmit range.

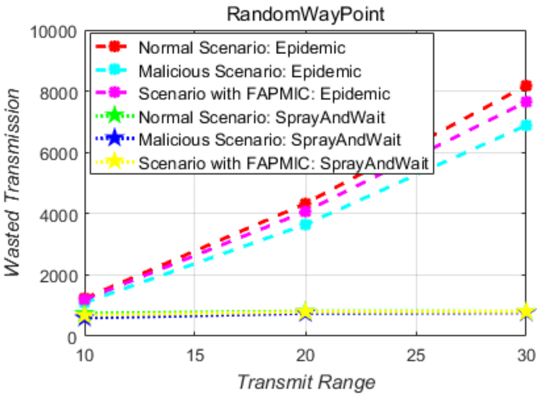


FIGURE 12. Average wasted transmission with transmit range. RWP model.

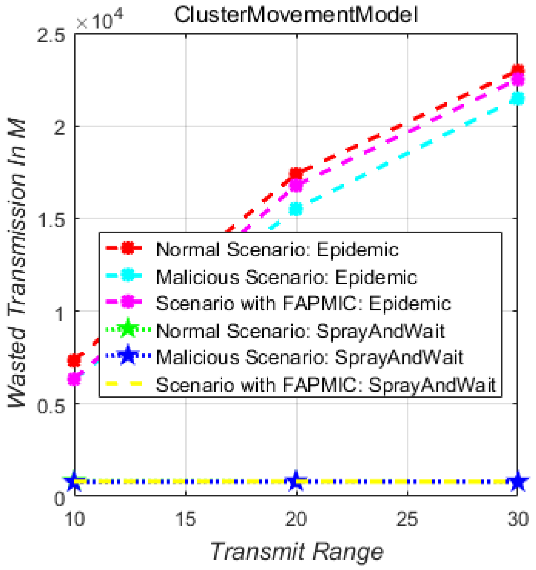


FIGURE 13. Average wasted transmission with transmit range. CMM model.

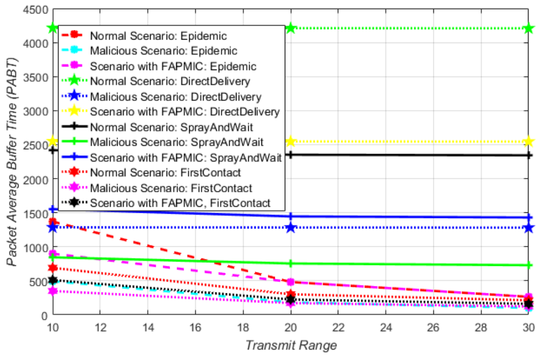


FIGURE 14. PABT of various protocols with transmit range. RWP model.

simulation results of Wasted-Transmission (Bandwidth-Consumption, WT/BC) of routing protocols with RWM, RWP and CMM respectively. Simulation results clearly indicate that WT is very high in the case of a normal scenario. Because in the normal scenario (scenario without malicious nodes) there are no malicious nodes, all nodes are benign which forward a lot of genuine packets, which is why it consumes a lot of bandwidth. When the malicious nodes launch attacks, they drop some packets which are subtracted from the relayed packets. That is why it consumed a small amount of bandwidth than the normal scenario. Simulation results clearly show a higher bandwidth consumption in our proposed algorithm than in malicious scenarios (Because the proposed algorithm blacklist the malicious nodes which launch the SPDA). When WT is high it implies some certain malicious nodes are blacklisted, which improve the ratios of PDR, this ultimately implies a higher bandwidth consumption.

Fig. 14, Fig. 15 and Fig. 16 shows simulation results of packet-average-buffer-time (PABT) of various routing protocols with RWM, RWP and CMM respectively.

Simulation results show that PABT is high when there are no malicious nodes. When some malicious nodes launch the SPDA, so drop ratios become high which is why some packets do not reach their destination. This spends fewer times in the buffer, which is why PABT is decreased in malicious scenarios. Testing results clearly demonstrate that

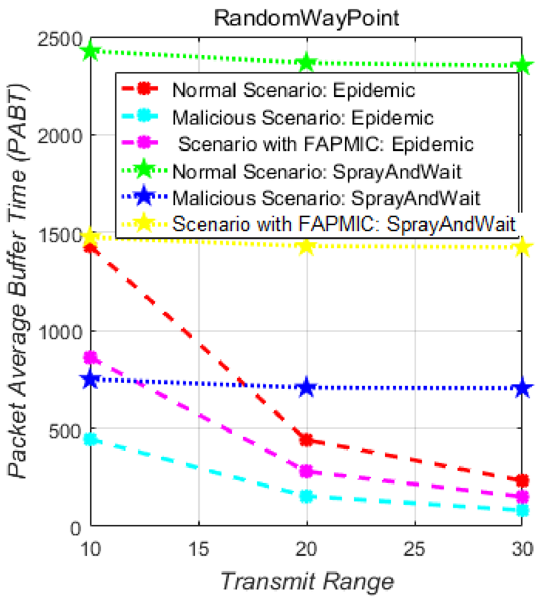


FIGURE 15. PABT of various routing protocols with various transmit range.

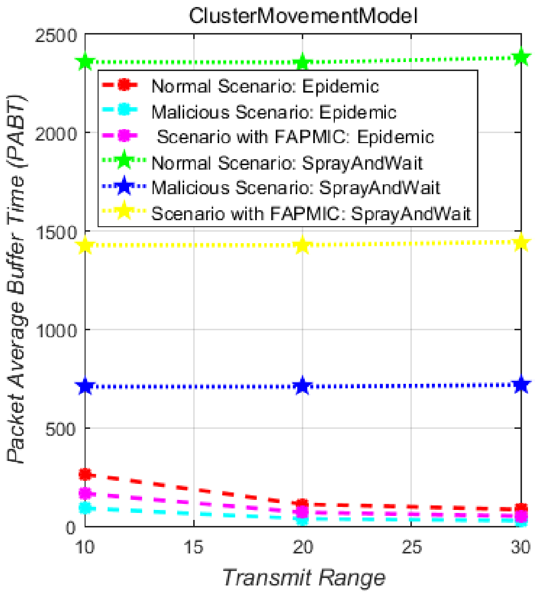


FIGURE 16. PABT of various routing protocols with various transmit range.

the proposed algorithm FAPMIC improved PABT with all routing protocols. Because FAPMIC detects misbehavior nodes and stops the SPDA to some certain extent. That is why packets spend more time in the buffer. Simulation results clearly show that PABT with RWM is high for RWP and CMM. Because in RWP models, nodes randomly move in all direction that is why the probability of the number-of-encounter with the nodes are high, which improve PDR and PABT. PABT of CMM is a little bit less than RWP because in CMM transmission areas are very small (small clusters), so it sends a lot of packets (due to buffer overloading packets being dropped in CMM). Simulation results also clearly proved that the PABT of SparyAndWait is high relative to other routing protocols (The EpidemicRouter continuously floods packets which is why PABT is less). Because SprayAndWait sprays and then waits sometimes that is why the PABT of the SprayAndWait is high.

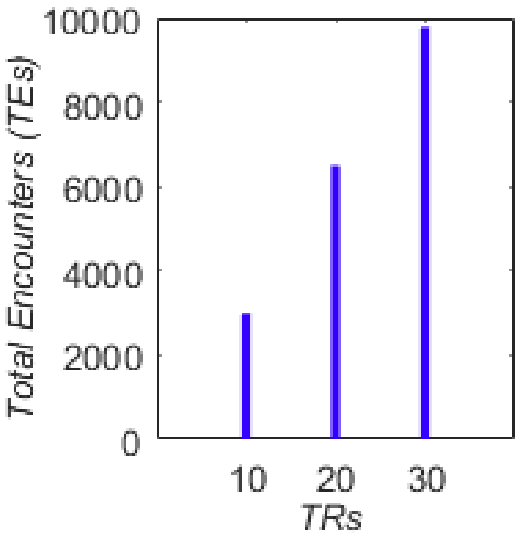


FIGURE 17. Total encounter (TEs) with transmit range.

Fig. 17 shows testing results of total-encounter/contact (TEs) with various transmit ranges (For the demonstration purpose this paper only mentioned results of RWM (RWP and CMM shows similar results)). Simulation results clearly show that TEs are increased with transmission range. This ultimately enhanced PDR (due to high numbers of encounters) and attack detection probability (This is the prof of the claim of this article, which is already mentioned in the Mathematical Evaluation section of this article).

Fig. 18, Fig. 18 and Fig. 20 illustrate the testing results of TEs with simulation-time with RWM, RWP, and CMM respectively. Experimental results demonstrate that TEs are higher in CMM relative to RWM and RWP (the reason for this is already mentioned in this article). Simulation results clearly proved that TEs are increased with TRs.

From simulation results this article concluded that due to SPDA, PDR and AL is decreased while PLR is increased. Moreover, PABT is decreased due to the SPDA. It is also observed from simulation results that RWP is mostly affected while CMM is least affected in term of PDR. This article also concluded from simulation results that TEs is increases with transmission range, which further improve PDR.

2) EXPERIMENT 02

In "experiment 02" this article calculated the DAC, detection delay, and false positive/false negative ratios with various strategies (tests) of the FAPMIC. Fig. 21 (TRs=10) and Fig. 22 (TRs=20) show simulation results of the DAC of the SPDA and the FPA of routing protocols with a number of packets (Just for the demonstration, this paper only

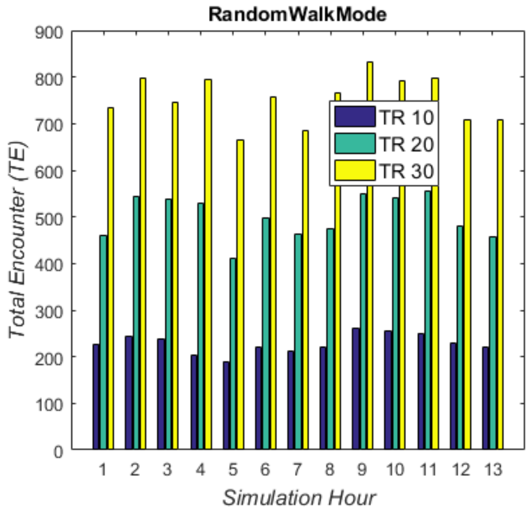


FIGURE 18. Total encounter (TEs) with simulation time. RWM.

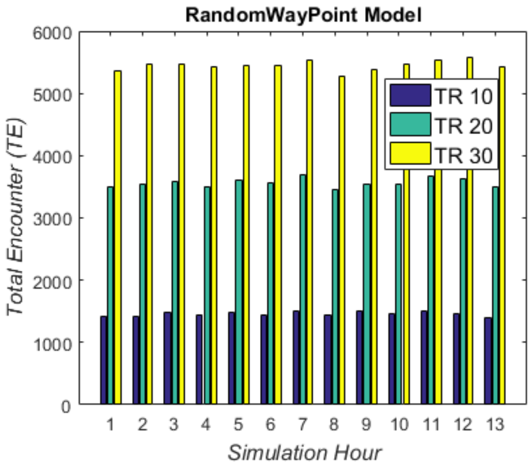


FIGURE 19. Total encounter with simulation time.

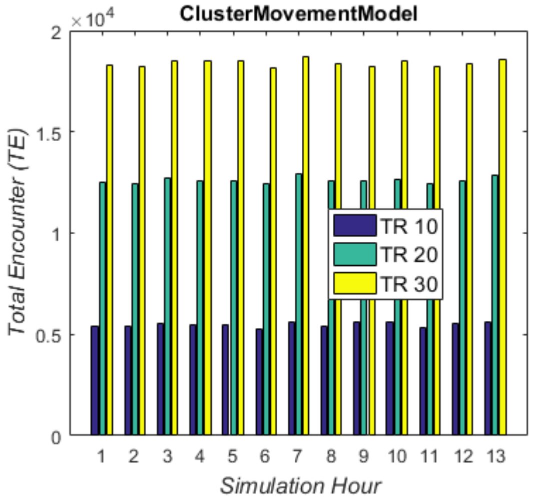


FIGURE 20. Total encounter with simulation time.

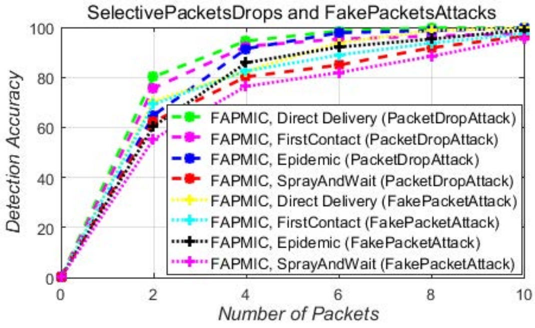


FIGURE 21. Detection accuracy of FAPMIC of routing protocols with packets.

shows experimental results of RandomWayPoint mobility model, however, ClusterMovement model further improved the DAC, due to a high number of encounters, this fact is mentioned already in this article in Fig. 20). Simulation results show that the DAC of DirectDelivery is high relative to other protocols. This is because DirectDelivery forwards packets directly to a final destination, which obviously enhances the DAC. The DAC of the FirstContact is little bit below than the DirectDelivery because the FirstContact forwards messages to the first contacted nodes. Actually, messages are delivered with at least two hops (that is why accuracy is low). The DAC of the SprayAndWait is minimum, because the SprayAndWait wait some certain times after the spray phase, which obviously takes some time to deliver the messages to the destination (Attacks are not detected in the FAPMIC when at least one packet and all packets did not reach the destination in the SPDA and the FPA respectively). The DAC of the EpidemicRouter is a little bit higher than the SprayAndWait, because the EpidemicRouter flood packets, due to this flooding, the probability of the packets' delivery to the destination is enhanced, which further enhances the the DAC.

Testing results illustrate that the DAC of the SPDA is a bit higher than the FPA. Because the SPDA is detected in the the FAPMIC when at least one packet reaches its destination unlike the FPA detection (FPA detected only when all packets are delivered to the destination). The experimental results clearly illustrate that the DAC is enhanced with the TRs (Almost 5 to 8 percent improvement in the DAC from TRs=10 to TRs=20). Because it is already stated in this article that the TRs are directly proportional to TEs, PDR increases with the TEs, this further improves the DAC.

Fig. 23 illustrates simulation results of the detection delay of the SPDA and the FPA of routing protocols with the NPs. Simulation results clearly demonstrate that the detection delay of the DirectDelivery is high among all simulated protocols. The DirectDelivery forwards packets directly to the final destination, which obviously consumes long times (because of disruption/disconnectivity, that is why the detection delay is high). The Detection delay of the FirstContact is a little bit below than DirectDelivery because

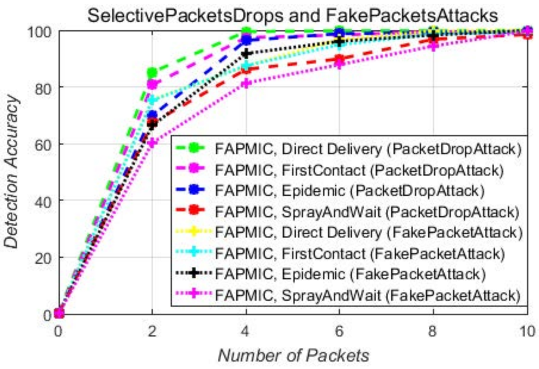


FIGURE 22. Detection accuracy of FAPMIC of routing protocols with packets.

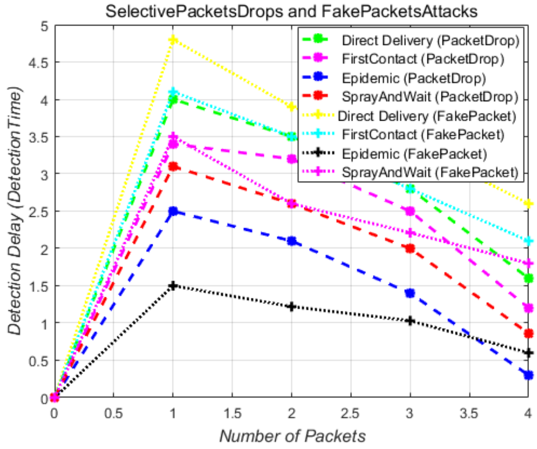


FIGURE 23. Detection delay of FAPMIC with number of packets.

the FirstContact sends packets to the first encounter node (packets reach the destination in at least two hops). Thus, packet delivery to the destination is the responsibility of that particular first contacted node (Due to the sparse nature of DTNs, it takes significant time for the packets to be delivered to the destination). The detection delay of the SprayAndWait is moderate because SprayAndWait waits significant times after the spray phase. The detection delay of the Epidemic is a little bit lesser than the SprayAndWait because the EpidemicRouter flood packets. The EpidemicRouter forwards more packets among all simulated protocols, which enhances TEs, TEs further improve PDR, and this further improved the DAC and detection delay. Furthermore, the results clearly illustrated, the detection delay of the SPDA is a little bit higher than the FPA (the reasons are already mentioned in the simulation results of the DAC).

Fig. 24 demonstrates the false positive ratios in our proposed algorithm FAPMIC for various routing protocols with NPs. The simulation results clearly indicated that the ratios of the false positives are sequentially decreasing with the NPs. The simulation results demonstrate that the false positive of the DirectDelivery (packet drops ratios are low in DirectDelivery) is minimum and the Epidemic (packets drops ratios are high in Epidemic) is higher among all simulated routing protocols.

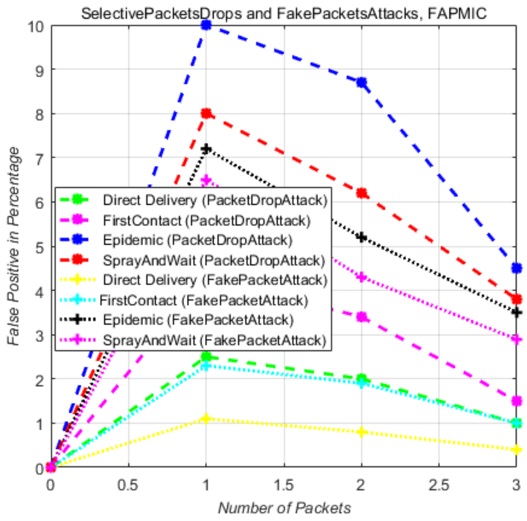


FIGURE 24. False positive of FAPMIC with number of packets.

The reason behind false positive ratios in our proposed algorithm FAPMIC is due to the packet drops (drops due to some other reasons (buffer overloading), not because of the malicious nodes). Considers an attack scenario in which malicious nodes drop selective packets. When the destination nodes verify the root-hash, if the root-hash does not verify, so the destination report that particular node to the TA. The TA collects EHs information and counts the number of packets. If the forwarding packets are not equal to receiving packets, then TA finds those malicious nodes from the EHs that drop packets. Considers an attack scenario, the node "A" forwards packet to the "C", and the "C" forwards that packet to the "D" (final destination is the "E"), the "D" drops that packet, but the packet is dropped due to some other reasons (memory overloading, or something else). In this case the TA blacklist the node "D" (reports the node "D" is malicious), however, in the reality, the node "D" is not malicious (false positive, but this ratio is significantly reduced with a number of received packets which depend on TRs, node processing capability and buffer management). Simulation results clearly show that false positive ratios of the FPA is a little bit less than the SPDA (fewer chances of false positive because algorithm detects the FPA only when all packets reach their destination). Considers a second attack scenario, in which malicious nodes (one node drops and one node inject a new fake packet) drop and inject a fake packet. When the destination nodes compare the root-hash, which obviously is not matched. After reporting to the TA (TA runs in the previously mentioned detection process). The TA blacklists packet dropping malicious node and leave fake packet attacker node (algorithm run in this manner. False negative). However, these types of attacks happened very rarely. The false negative ratios of our proposed algorithm is almost zero (almost negligible in the simulation results) with all routing protocols, that is why this article not shown results here to just save space (this article shows simulation results of false negative rate in comparison with previously proposed algorithms in the comparison section).

From simulation results this article concluded that DAC of DirectDelivery is high and minimum in SprayAndWait. From simulation results this article observed that DAC is enhance with TRs. From simulation results this article concluded that detection delay of DirectDelivery is high among all routing protocols. Moreover, the experimental results indicates that the ratios of false positive is decreasing with NPs. It is also observed from simulation results that the ratios of false positive of DirectDelivery is low while Epidemic have high.

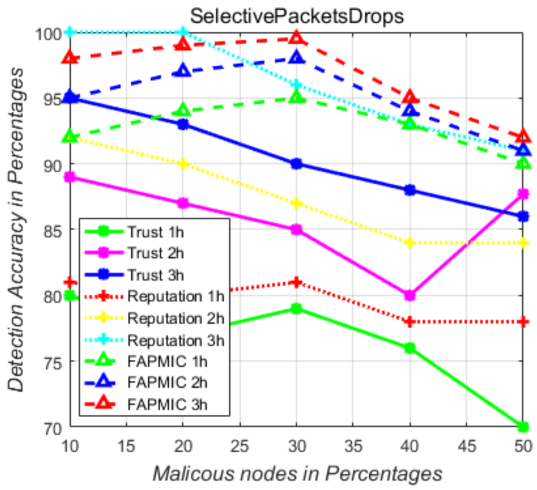


FIGURE 25. Detection accuracy with various percentages of malicious nodes.

VII. COMPARISON

A. DETECTION ACCURACY

This article compared various research articles (with various tests) with our proposed algorithm FAPMIC for the DAC. Fig. 25 shows simulation results of the DAC of previously proposed article [48] (Reputation) and the article [46] (Trust) with FAPMIC (With same simulation setup [48] and [46]). Testing results clearly illustrate that the DAC decreases with the number of malicious nodes in [48] and [46] with simulation times. The simulation results demonstrate the DAC is improved with simulation times. The simulation results of our proposed algorithm FAPMIC clearly shows that the DAC is initially enhanced with some malicious nodes and then starts to decline. This is because, our proposed algorithm detects malicious nodes when the TA collects EHs from all nodes. When the number of misbehaving malicious/selfish nodes are increased, the probability of the packet collections (NON is directly proportional to TEs, which further enhanced the PDR and the DAC) are increased which improve the DAC. However, when the number of malicious nodes cross some specific limit (number of malicious nodes increase) so the DAC starts to decline because the EHs collection capability of the TA starts to decline (due to the memory overloading, some packets drop, the packets processing capability of the TA declined, so therefore the TA needs some certain time to collects all EHs packets). The simulation results indicate that the DAC of our proposed scheme is better than [48] and [46].

Fig. 26 shows comparison results of the FAPMIC with previously proposed article [51] (PIDMIO) for the FPA DAC (With the same simulation setup as PIDMIO, TRs 10). The simulation results of PIDMIO clearly shows up-and-down graph (like a zigzag path) when the number of malicious nodes are increased (no clear indication that the DAC either decreases or increases with intruder nodes). The simulation results clearly demonstrate that our proposed algorithm FAPMIC improved the DAC (initially below/lesser than PIDMIO because in our scheme, the TA collects EHs packets which depends on encounters, then after some time the graph becomes stable when the TA collects EHs from various nodes) as compared to PIDMIO. It is also clear from experimental results that the DAC is increases with the number of malicious nodes (this article already mentioned up to some certain limit the DAC are increased then starts to decreases).

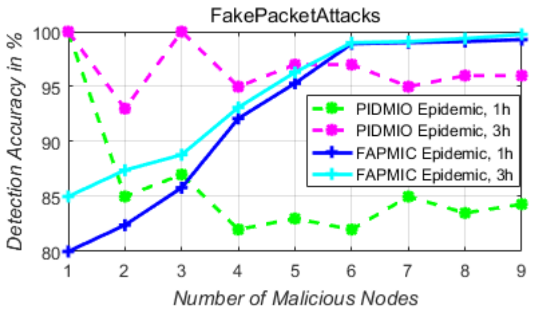


FIGURE 26. Detection accuracy with number of malicious nodes.

Fig. 27 shows comparison results of FAPMIC and article [23] (DAPCA) for the FPA DAC (With the same simulation setup like DAPCA). Testing results clearly demonstrate that DAC is decreases with malicious nodes in the DAPCA. From experimental results this research works concluded that the DAC of our proposed algorithm FAPMIC is increases with number of malicious nodes then starts to decreases (reasons are already mentioned in this article). Simulation results clearly indicate that our proposed algorithm FAPMIC performs better than the DAPCA in EpidemicRouter and FirstContact (DAPCA, DAC of SprayAndWait is better than FAPMIC).

B. FALSE POSITIVE AND FALSE NEGATIVE

This article compared simulation results of false positive rate (the SPDA) of article [46] (Trust) with our proposed algorithm FAPMIC (with same simulation setup, this article shows results of Epidemic). Article [46] only shows results of false positives (that is why this article compared results of FAPMIC only with article [46]). The simulation results show that, false positive rate is increased (Unlike in the Trust-Based-Algorithm in which false positive rate is

TABLE 4. Comparison table.

Article	FPAD	SPDAD	PDR/PLR	AL	WT	NE	PABT	DAC	DD	FP	FN	PR
[49]	Yes	No	-	-	-	-	-	+	-	-	+	+
[21]	Yes	No	-	-	-	-	-	+	-	-	-	+
[22]	Yes	No	-	-	-	-	-	+	-	-	+	-
[23]	Yes	No	+	-	-	-	-	+	-	+	-	+
[46]	No	Yes	-	-	-	-	-	+	-	+	-	-
[47]	No	Yes	-	-	-	-	-	+	-	-	+	-
[51]	Yes	No	+	-	-	-	-	+	-	+	-	-
[48]	No	Yes	+	-	-	-	-	+	-	-	-	-
FAPMIC	Yes	Yes	+	+	+	+	+	+	+	+	+	-

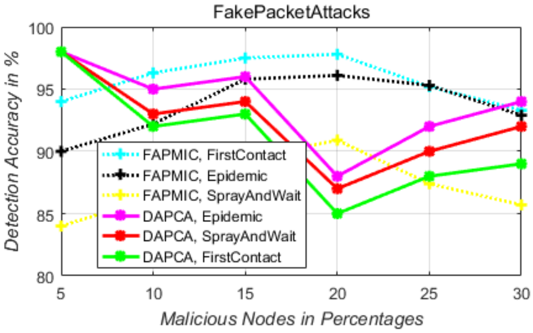


FIGURE 27. Detection accuracy with malicious nodes in percentage.

decreased with misbehavior nodes) with misbehavior nodes in the FAPMIC (Reasons for this are already mentioned in the simulation results of this article in Experiment 02). The simulation results clearly indicate that the ratios of false positives of the FAPMIC is better than article [46] when the number of malicious nodes are below seventy percent (seventy percent is a very high percentage, the probability of malicious nodes above seventy percent is very rare (low)).

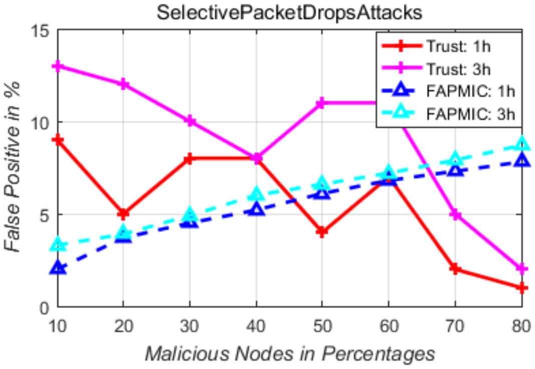


FIGURE 28. False positive rate.

Fig. 29 shows simulation results of false positive and false negative rate (the FPA) of the FAPMIC in comparison with the DAPCA and the article [49] (PFA) respectively (the same setup with DAPCA and PFA, this paper demonstrates testing results of only EpidemicRouter for comparison purposes, the false positive rate of other routing protocols are better than EpidemicRouter). The experimental results clearly indicate that the rate of the false positives increases with the number of malicious nodes (reasons are already outlined in this article). The simulation results clearly demonstrate that our proposed algorithm FAPMIC reduced/enhanced the number of false positives relative to the DAPCA. The simulation results also show the false negative rate of the PFA relative to the FAPMIC. The simulation results clearly show that the false negative rates increases with the number of malicious nodes in the PFA. The simulation results also demonstrate that the false negative ratios of our proposed algorithm FAPMIC is almost zero (Reasons for this are already mentioned in the simulation results of Experiment 02).

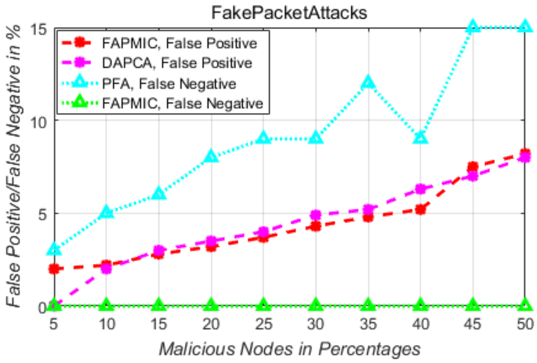


FIGURE 29. False positive and false negative rate.

C. MISCELLANEOUS PARAMETERS

This research work demonstrates various parameters in the simulation and results section of this article. The simulation results show that our proposed algorithm FAPMIC enhanced resources consumption, this enhancement further improved the PDR, PLR, PABT, WT, AL, and TEs. The experimental results of the aforementioned parameters are not shown in the previously proposed articles. That is why this paper does not compared the aforementioned parameters (not possible to compare). However, this is already illustrated in the simulation section of this article, that the algorithm FAPMIC improved the aforementioned parameters. Table 4 summarizes the achievements/contributions of this article relative to the previously proposed algorithms in this security domain. In Table 4 FPAD is the fake packet attack detection, SPDAD is the selective packet drop attack detection, DD is the detection delay, FP is the false positive, FN is the false negative, NE is the number-of-encounters and PR is the probabilistic results (+ in the table means researchers show that particular parameter in the paper and - means researchers did not show that particular parameter in the paper).

VIII. CONCLUSION AND FUTURE WORKS

The focus of this research works on misbehavior node mitigation in ICNs, which launch various attacks. Moreover, these attacks further degrade/demolish the network performance. The malicious nodes often launch various attacks, the goal of the misbehavior nodes are to mainly drop packets or inject bogus-packets/fake-packets to degrade the network operations. We further concluded from the simulation results that due to the aforementioned attacks, delivery ratio decreased while packet loss ratios increased. Also, the misbehavior nodes overused scarce resources (consumed buffer, bandwidth), created the nodes unavailability, and disseminated the fake packets. This article presented an algorithm FAPMIC, which mitigates the SPDA and the FPA. The simulation results clearly demonstrate that an algorithm FAPMIC mitigates misbehavior malicious/selfish nodes, and save limited resources of the DTNs. This further improved the delivery ratios, loss ratios, DD, DAC, and reduced the FP, and the FN rates. This paper concluded from the experimental results that the TRs are directly proportional to the number-of-encounter (contacts), which further improved the PDR, DAC, DD, FP, and the FN ratios.

From the simulation results this article concluded that due the SPDA, all previously proposed routing protocols are affected, however, the SprayAndWait are mostly and the FirstContact is least affected. This is because our proposed algorithm detects attacks when the nodes share EHs packets with the TA. SprayAndWait waits some certain times after the spray phase, that is why in this idle time the proposed algorithm does not detects the malicious nodes. FirstContact forwards packets to only the first contacted node (the packet is delivered to the destination with at least two hops), so the probability of the detection is high, which is why the attacks have minimum effect on the FirstContact. This paper also analyses from testing results that the DAC is enhanced with the NPs (FAPMIC detects attacks when the nodes share more EHs packets), also the DAC is increased with the NON up to some certain limit (NON is directly proportional to the NPs) then starts to decreased (because when malicious nodes cross some certain limit, which sends more packets and the TA cannot collects all packets, that is why the DAC starts to decline). From the simulation results, this article observes the rate of FP is decreased with the NPs and increased with the number-of-malicious nodes.

This article also launches various theoretical attacks on previously proposed algorithms/FAPMIC. This cryptanalysis clearly show loopholes on previously proposed algorithms and FAPMIC as well. These analyses (mathematical analyses and cryptanalysis of this article) hopefully will improve the design of the detection algorithms in the future.

Hopefully, this article will further motivates the researchers' interest in this security domain and further highlight the following directions for investigation.

* Proposed an algorithm/method which detects a particular misbehavior node, which launches the SPDA and the FPA at the same time.
* Proposed distributed-based detection algorithm which overcomes a single point of failure of the centralized-based detection algorithms.
* Exact relationship (quantitative-based analyses/ simulation-based analyses) of the parameters and constants used in the mathematical evaluation section of this article.
* Artificial-intelligence-based algorithm that detects bogus bundles and intruder nodes that inject bogus bundles into the networks.
* Mathematical system which track the position of all nodes in vehicular networks.

ACKNOWLEDGMENT

The authors would like to thank the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication and also would like to thank the valuable suggestions and comments of Dr. Zahid Ullah of the Department of Computer Science, Institute of Management Sciences Peshawar, Pakistan, in this article.

CONFLICT OF INTEREST

The authors declared that there is no conflict of interest among any author.

REFERENCES

[1] L. Wu, S. Cao, Y. Chen, J. Cui, and Y. Chang, "An adaptive multiple spray-and-wait routing algorithm based on social circles in delay tolerant networks," *Comput. Netw.*, vol. 189, Apr. 2021, Art. no. 107901.

[2] C. Caini, "Delay-tolerant networks (DTNs) for satellite communications," in *Advances in Delay-Tolerant Networks (DTNs)*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 23–46.

[3] A. Altaweel, R. Stoleru, G. Gu, A. K. Maity, and S. Bhunia, "On detecting route hijacking attack in opportunistic mobile networks," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 24, 2022, doi: 10.1109/TDSC.2022.3186029.

[4] C. Choudhari and D. Niture, "Disruption tolerant network (DTN) for space communication: An overview," in *Proc. IEEE 7th Int. Conf. Converg. Technol. (ICT)*, Apr. 2022, pp. 1–5.

[5] Y. Azzoug and A. Boukra, "Enhanced UAV-aided vehicular delay tolerant network (VDTN) routing for urban environment using a bio-inspired approach," *Ad Hoc Netw.*, vol. 133, Aug. 2022, Art. no. 102902.

[6] K. Matsuo, E. Kulla, and L. Barolli, "Effect of transporter autonomous underwater vehicles for underwater optical wireless communication considering delay tolerant networks," in *Proc. Int. Conf. Network-Based Inf. Syst.* Cham, Switzerland: Springer, 2022, pp. 172–181.

[7] S. Perumal, V. Raman, G. N. Samy, B. Shanmugam, K. Kisenasamy, and S. Ponnan, "Comprehensive literature review on delay tolerant network (DTN) framework for improving the efficiency of internet connection in rural regions of Malaysia," *Int. J. Syst. Assurance Eng. Manag.*, vol. 13, pp. 764–777, Jan. 2022.

[8] W. Khalid, N. Ahmed, M. Khalid, A. U. Din, A. Khan, and M. Arshad, "FRID: Flood attack mitigation using resources efficient intrusion detection techniques in delay tolerant networks," *IEEE Access*, vol. 7, pp. 83740–83760, 2019.

[9] W. Khalid, Z. Ullah, N. Ahmed, Y. Cao, M. Khalid, M. Arshad, F. Ahmad, and H. Cruickshank, "A taxonomy on misbehaving nodes in delay tolerant networks," *Comput. Secur.*, vol. 77, pp. 442–471, Aug. 2018.

[10] C. Chakrabarti and S. Pramanick, "Implementing data security in delay tolerant network in post-disaster management," in *Computational Advancement in Communication, Circuits and Systems* (Lecture Notes in Electrical Engineering). Berlin, Germany: Springer, 2022, pp. 77–92.

[11] L. Cao and R. Viswanathan, "Average operation time of bundle protocol in delay/disruption-tolerant networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 8, pp. 5801–5813, Aug. 2022.

[12] A. Mallorqui, A. Zaballos, and D. Serra, "A delay tolerant network for Antarctica," *IEEE Commun. Mag.*, vol. 60, no. 12, pp. 56–62, Dec. 2022.

[13] S. Chatterjee, M. Nandan, A. Ghosh, and S. Banik, "DTNMA: Identifying routing attacks in delay-tolerant network," in *Cyber Intelligence and Information Retrieval* (Lecture Notes in Electrical Engineering). Berlin, Germany: Springer, 2022, pp. 3–15.

[14] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," *Arabian J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, 2017.

[15] R. Sharma and S. K. Dinkar, "Selfish node detection by modularized deep NMF autoencoder based incentivized reputation scheme," *Cybern. Syst.*, vol. 47, pp. 1–27, Jun. 2022.

[16] M. Humayun, M. Niazi, N. Jhanjhi, M. Alshayeb, and S. Mahmood, "Cyber security threats and vulnerabilities: A systematic mapping study," *Arabian J. Sci. Eng.*, vol. 45, no. 4, pp. 3171–3189, Apr. 2020.

[17] A. Bang and U. P. Rao, "Impact analysis of rank attack on RPL-based 6LoWPAN networks in Internet of Things and aftermaths," *Arabian J. Sci. Eng.*, vol. 47, pp. 1–17, Oct. 2022.

[18] D. Rangwani and H. Om, "A robust four-factor authentication protocol for resource mining," *Arabian J. Sci. Eng.*, vol. 47, pp. 1–25, Jul. 2022.

[19] B. V. Sherif and P. Salini, "Selfish node management in opportunistic mobile networks with improved social based watchdog and dynamic power AODV protocol," *Int. J. Inf. Technol.*, vol. 14, no. 6, pp. 3253–3264, Oct. 2022.

[20] F. R. C. Araujo, A. L. R. Madureira, and L. N. Sampaio, "A multicriteria-based forwarding strategy for interest flooding mitigation on named data wireless networking," *IEEE Trans. Mobile Comput.*, early access, Sep. 12, 2022, doi: 10.1109/TMC.2022.3206167.

[21] M. Alajeely, R. Doss, and V. Mak-Hau, "Catabolism attack and anabolism defense: A novel attack and traceback mechanism in opportunistic networks," *Comput. Commun.*, vol. 71, pp. 111–118, Nov. 2015.

[22] M. Alajeely, A. Ahmad, and R. Doss, "Malicious node traceback in opportunistic networks using Merkle trees," in *Proc. IEEE Int. Conf. Data Sci. Data Intensive Syst.*, Dec. 2015, pp. 147–152.

[23] M. Alajeely, R. Doss, A. Ahmad, and V. Mak-Hau, "Defense against packet collusion attacks in opportunistic networks," *Comput. Secur.*, vol. 65, pp. 269–282, Mar. 2017.

[24] J. Khochare, C. Joshi, B. Yenarkar, S. Suratkar, and F. Kazi, "A deep learning framework for audio deepfake detection," *Arabian J. Sci. Eng.*, vol. 47, no. 3, pp. 3447–3458, Mar. 2022.

[25] M. Arshad, Z. Ullah, M. Khalid, N. Ahmad, M. Khalid, D. Shahwar, and Y. Cao, "Beacon trust management system and fake data detection in vehicular ad-hoc networks," *IET Intell. Transp. Syst.*, vol. 13, no. 5, pp. 780–788, 2018.

[26] S. A. M. Benazir and V. Umarani, "Detection of selfish & malicious behavior using DTN-chord monitoring in mobile networks," in *Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES)*, Feb. 2016, pp. 1–5.

[27] M. Khalid, Y. Cao, N. Ahmad, W. Khalid, and P. Dhawankar, "Radius-based multipath courier node routing protocol for acoustic communications," *IET Wireless Sensor Syst.*, vol. 8, no. 4, pp. 183–189, Aug. 2018.

[28] K. Nitesh, S. Malwe, and A. K. Keshari, "Efficient trajectory formulation for drone sink in wireless sensor networks: An asanoha-based approach," *Arabian J. Sci. Eng.*, vol. 47, pp. 10071–10084, Jan. 2022.

[29] M. Khalid, Y. Cao, N. Aslam, C. Suthaputchakun, M. Arshad, and W. Khalid, "Optimized pricing & scheduling model for long range autonomous valet parking," in *Proc. Int. Conf. Frontiers Inf. Technol. (FIT)*, Dec. 2018, pp. 65–70.

[30] F. Cadet and D. T. Fokum, "Coping with denial-of-service attacks on the IP telephony system," in *Proc. SoutheastCon*, Mar. 2016, pp. 1–7.

[31] Y. Shimizu, T. Kimura, and J. Cheng, "Performance evaluation of a hash-based countermeasure against fake message attacks in sparse mobile ad hoc networks," *IEICE Trans. Commun.*, vol. 105, no. 7, pp. 833–847, 2022.

[32] Z. Gao, H. Zhu, S. Du, C. Xiao, and R. Lu, "PMDS: A probabilistic misbehavior detection scheme in DTN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2012, pp. 4970–4974.

[33] C. Chakrabarti, A. Banerjee, and S. Roy, "An observer-based distributed scheme for selfish-node detection in a post-disaster communication environment using delay tolerant network," in *Proc. Appl. Innov. Mobile Comput. (AIMoC)*, 2014, pp. 151–156.

[34] C. Chakrabarti, S. Chakrabarti, and A. Banerjee, "A dynamic two hops reputation assignment scheme for selfish node detection and avoidance in delay tolerant network," in *Proc. IEEE Int. Conf. Res. Comput. Intell. Commun. Netw. (ICRCICN)*, Nov. 2015, pp. 345–350.

[35] Q. Li and G. Cao, "Mitigating routing misbehavior in disruption tolerant networks," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 664–675, Apr. 2012.

[36] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen, "SMART: A secure multilayer credit-based incentive scheme for delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 4628–4639, Oct. 2009.

[37] H. Zhu, X. Lin, R. Lu, and X. Shen, "A secure incentive scheme for delay tolerant networks," in *Proc. 3rd Int. Conf. Commun. Netw. China*, 2008, pp. 23–28.

[38] K. Devi and P. Damodharan, "Detecting misbehavior routing and attacks in disruption tolerant network using itrm," in *Proc. Int. Conf. Current Trends Eng. Technol. (ICCTET)*, Jul. 2013, pp. 334–337.

[39] E. Ayday and F. Fekri, "An iterative algorithm for trust management and adversary detection for delay-tolerant networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 9, pp. 1514–1531, Sep. 2012.

[40] A. A. Chandavale and T. P. Chaure, "An approach to detect malicious node for delay tolerant networks," in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2015, pp. 1–6.

[41] L. Kulkarni, D. Mukhopadhyay, and J. Bakal, "Analyzing security schemes in delay tolerant networks," in *Proc. Int. Conf. Data Eng. Commun. Technol.* Cham, Switzerland: Springer, 2017, pp. 613–620.

[42] M. Malathi and S. Jayashri, "Design and performance of dynamic trust management for secure routing protocol," in *Proc. IEEE Int. Conf. Adv. Comput. Appl. (ICACA)*, Oct. 2016, pp. 121–124.

[43] R.-I. Ciobanu, C. Dobre, M. Dascalu, S. Trausan-Matu, and V. Cristea, "Collaborative selfish node detection with an incentive mechanism for opportunistic networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, May 2013, pp. 1161–1166.

[44] P. Asuquo, H. Cruickshank, C. P. A. Ogah, A. Lei, and Z. Sun, "A collaborative trust management scheme for emergency communication using delay tolerant networks," in *Proc. 8th Adv. Satell. Multimedia Syst. Conf. 14th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, Sep. 2016, pp. 1–6.

[45] S. Basu and S. Roy, "A global reputation estimation and analysis technique for detection of malicious nodes in a post-disaster communication environment," in *Proc. Appl. Innov. Mobile Comput. (AIMoC)*, Feb. 2014, pp. 179–185.

[46] A. Ahmad, M. Alajeely, and R. Doss, "Establishing trust relationships in OppNets using Merkle trees," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2016, pp. 1–6.

[47] A. Ahmad, M. Alajeely, and R. Doss, "Defense against packet dropping attacks in opportunistic networks," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2014, pp. 1608–1613.

[48] A. Ahmad, M. Alajeely, and R. Doss, "Reputation based malicious node detection in OppNets," in *Proc. 13th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2016, pp. 1–6.

[49] M. Alajeely, A. Ahmad, R. Doss, and V. Mak-Hau, "Packet faking attack: A novel attack and detection mechanism in OppNets," in *Proc. 10th Int. Conf. Comput. Intell. Secur.*, Nov. 2014, pp. 638–642.

[50] R. Wazirali, "An improved intrusion detection system based on KNN hyperparameter tuning and cross-validation," *Arabian J. Sci. Eng.*, vol. 45, no. 12, pp. 10859–10873, Dec. 2020.

[51] A. Ahmad, R. Doss, M. Alajeely, S. F. Al Rubeaai, and D. Ahmad, "Packet integrity defense mechanism in OppNets," *Comput. Secur.*, vol. 74, pp. 71–93, May 2018.

[52] T. N. D. Pham, C. K. Yeo, N. Yanai, and T. Fujiwara, "Detecting flooding attack and accommodating burst traffic in delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 795–808, Jan. 2018.

[53] Q. Li, W. Gao, S. Zhu, and G. Cao, "To lie or to comply: Defending against flood attacks in disruption tolerant networks," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 3, pp. 168–182, May/Jun. 2013.

[54] A. Keränen, J. Ott, and T. Kärkkäinen, "The ONE simulator for DTN protocol evaluation," in *Proc. 2nd Int. ICST Conf. Simul. Tools Techn.*, 2009, p. 55.

WAQAR KHALID received the bachelor's degree (Hons.) in computer science from the Institute of Business and Management Study (IBMS), The University of Agriculture, Peshawar, Khyber Pakhtunkhwa, Pakistan, and the Master of Science (M.S./M.Phil.) degree in computer science from the Institute of Management Sciences (IM-Sciences) Peshawar, Khyber Pakhtunkhwa. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Engineering, Wuhan University, China. He has been working as a Teacher with Elementary and Secondary Education, Khyber Pakhtunkhwa, since 2017. He is also working as a reviewer in various journals of IEEE and Elsevier. His research interests include the design and analyses of secured communication protocols in networks (self organizing networks: DTNs, VANETs, WSNs, SINs, and the IoTs). He design new cryptographic algorithms and cryptanalysis of the existing algorithms.

NAVEED AHMAD (Member, IEEE) received the Graduate degree in computer science from the Department of Computer Science, University of Peshawar, Pakistan, in 2007, and the Ph.D. degree in computer science from the Center for Communication System Research (CCSR), University of Surrey, U.K., in 2013. He is currently working as an Associate Professor at the College of Computer and Information Sciences, Prince Sultan University, Saudi Arabia. Prior to his current position, he remained as an Assistant Professor at the Department of Computer Science, University of Peshawar. He has over 45 publications that include international reputed journals, conferences, and workshops. He worked in the area of cyber security, privacy, blockchain technology, and penetration testing. He has managed three research and development grants related to blockchain, transport system for emergency vehicles, and platoon management. He remained as a reviewer of various research proposals/grants funded by the U.K. Research and Innovation (UKRI). He also served on the program committee for various national conferences and workshops. He remained as a reviewer of various IEEE, Elsevier, Springer, and MPDI journals. He has the honor of serving as the Guest Editor for a special issue in the IEEE INTERNET OF THINGS JOURNAL.

SULEMAN KHAN (Member, IEEE) received the Ph.D. degree (Hons.) in computer science and information technology from Universiti Malaya, Malaysia, in 2017. He was a Faculty Member of the School of Information Technology, Monash University, Malaysia, from June 2017 to March 2019. He was also a Faculty Member of the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, U.K. He is currently working as a Senior Lecturer at the University of Central Lancashire, U.K. He has published more than 80 high-impact research articles in reputed international journals and conferences. His research interests include network forensics, software-defined networks security, and the IoT security. He is also a PGCAP and a FHEA.

NAJAM U. SAQUIB received the master's degree in computer science from Abasyn University, Pakistan. He is currently pursuing the Ph.D. degree in computer science with the Capital University of Science & Technology, Islamabad, Pakistan. He works as a Web Developer at NS Bytes. His current research interests include multi-model-based emotion recognition in context to e-health and security attacks detection in *ad-hoc* networks.

MUHAMMAD ARSHAD received the Bachelor of Science degree (Hons.) in computer science from the Institute of Business and Management Sciences (IBMS), The University of Agriculture, Peshawar, Khyber Pakhtunkhwa, and the Master of Science degree in computer science from the Institute of Management Sciences (IMSciences) Peshawar, Khyber Pakhtunkhwa. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology, Beihang University, Beijing, China. His research interests include security in vehicular *ad-hoc* networks, security of space information networks, satellite communication, and blockchain.

DURI SHAHWAR received the M.S. and M.Phil. degrees in computer science from the Institute of Management Sciences (IM-Sciences) Peshawar, Khyber Pakhtunkhwa, Pakistan, in 2018. She is currently pursuing the Ph.D. degree with the National University of Computer and Emerging Sciences (FAST-NUCES), Peshawar, Pakistan. She is also working as a Lecturer with the Institute of Management Sciences (IM-Sciences) Peshawar. Her research interests include incorporating machine learning in large scale malware analysis and security in vehicular *ad-hoc* networks.