IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 21, 2018, accepted March 25, 2018, date of publication April 5, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2822945

Sensitivity Analysis of an Attack-Pattern Discovery Based Trusted Routing Scheme for Mobile Ad-Hoc Networks in Industrial IoT

RUTVIJ H. JHAVERI[1], NARENDRA M. PATEL[2], YUBIN ZHONG[3], AND ARUN KUMAR SANGAIAH[4]
[1]SVM Institute of Technology, Bharuch 392001, India
[2]Birla Vishvakarma Mahavidyalaya, V.V. Nagar 388120, India
[3]Guangzhou University, Guangzhou 510006, China
[4]Vellore Institute of Technology, Vellore 632014, India

Corresponding author: Yubin Zhong (zhong_yb@163.com)

ABSTRACT Mobile ad-hoc networks (MANETs) are pervasive autonomous networks that will play a vital role in future Industrial Internet-of-Things communication, where smart devices will be connected in a completely distributed manner. However, due to lack of infrastructure and absence of centralized administration, MANETs are shrouded with various security threats. Some internal mobile nodes in these resource constrained networks may compromise the routing mechanism in order to launch denial-of-service attacks to carry out distinct kinds of packet forwarding misbehaviors. In order to address this issue, in our previous paper, we devised a trusted routing scheme with pattern discovery (TRS-PD), which identifies packet dropping adversaries in advance by monitoring and analyzing their behavior during route discovery phase. In this paper, we perform sensitivity analysis of TRS-PD which is carried out by varying values of different parameters in distinct network scenarios in the existence of three distinct packet dropping attacks. In addition, this work summarizes the attack-pattern discovery mechanism, trust model, and routing mechanism adopted by TRS-PD in order to counter the adversaries which follow certain attack patterns along with other adversaries. Experiments conducted with network simulator-2 indicate the correct choices of parameter values for distinct network scenarios.

INDEX TERMS Attack-pattern discovery, industrial Internet-of-Things, mobile ad-hoc networks, packet forwarding misbehavior, sensitivity analysis, trusted routing.

I. INTRODUCTION

After the success of cellular and Wi-Fi technologies in the last two decades, wireless communication has become a popular way of communication in people's day to day life [2]. Ad-Hoc networks have grown in a thick and fast way as a result of increased need of eliminating fixed infrastructure, geographical dependence and complexity of deployment for critical applications such as Industrial IoT (IIoT), military operations, disaster relief management, maritime communications, intelligent transportation systems, wild-life monitoring, health monitoring and many more [2], [3], [4]. A Mobile Ad-hoc Network (MANET) is such an autonomous distributed network of mobile nodes which supports wireless communication in decentralized environment with geographical independence and impulsive deployment [4]–[6]. A mobile ad-hoc cloud is formed from such a MANET which inherits the merits of cloud computing paradigm such as flexibility, efficient resource utilization and enhanced manageability [7]–[9]. It provides services by exploiting the available computing resources in the mobile nodes [7].

At one end, where technologies are changing the digital world, security has still remained as a major concern for cyber security researchers [10], [11]. The inherent characteristics of MANETs along with resource constraints, dynamic topology and limited radio range bring new security challenges for IIoT applications [12] along with other challenges such as quality-of-service (QoS) improvement, optimal resource management, reliability and scalability [4], [13], [14].

Secure routing in ad-hoc networks has been one of the major concerns for researchers as the classical routing protocols for these networks assume cooperative trusted setting among the mobile devices [15], [16]. Legitimate internal mobile nodes can be easily compromised by attackers in order to perform various kinds of packet forwarding misbehaviors to launch DoS attacks. Sequence number attacks (such as blackhole and grayhole attacks) are such prominent DoS attacks against classical MANET routing protocols which may drop data packets during data transmission phase after breaking protocol rules during route discovery phase [17]. Therefore, it is imperative to prevent protocol malfunctioning at an early stage (during route discovery) to reduce further damage [18]. Trust-based secure routing has caught eyes of researchers in recent past for addressing various security issues [19], [20]. In our previous work, we proposed a trust-based scheme (TRS-PD) [1] which aims at detecting and isolating distinct kinds of packet dropping adversaries in advance.

To the best of our knowledge, TRS-PD is the first scheme to manage trust in MANETs which attempts to discover attack patterns followed by malevolent nodes during the route discovery process. This attack pattern discovery mechanism sits on the top of the trust model and employs the *Method of Common Differences* for the recorded field values of route reply packet to discover attack patterns. As a result, adversaries which follow certain attack patterns are identified by the attack pattern discovery mechanism while other packet dropping adversaries are identified by the trust model in this two-tier security scheme. The experimental results shown in [1] depict that TRS-PD leads to an earlier detection of packet dropping adversaries that follow specific patterns, which in turn increases packet delivery rate. In this paper, we perform sensitivity analysis of TRS-PD under different network conditions by varying values of distrust threshold, trust update interval and trust components' weights. This analysis provides the correct tuning of values of these trust parameters in different network settings. Sensitivity analysis of TRS-PD is carried out against three distinct adversary models described in [1] by varying different network parameters.

The remainder of the paper is organized as follows: In Section II, we discuss the existing trust-based routing schemes for ad-hoc networks. In Section III, we provide the discussion of our previous work, TRS-PD. In Section IV, we present the simulation results and analyze the performance of TRS-PD. Finally, the paper is concluded in Section V.

II. RELATED WORK

In this section, we discuss the relevant research works carried out to address the security requirements of ad-hoc routing protocols by way of trust management schemes.

Khan *et al.* [21] proposed a multi attribute trust framework (MATF), extended from optimized link state routing (OLSR) protocol, in order to enhance security in MANETs by improving the bootstrapping time, adversary detection rate, false positive rate and packet dropping rate. Instead of using single trust attribute, MATF uses multiple attributes such as control packet generation, control packet forwarding and data packet forwarding in order to expedite trust building process. Apart from using first-hand information, it uses second-hand information from watchdog nodes whose trust values are above threshold. The results depict effectiveness of MATF against packet dropping, packet modification and link withholding attacks. Gazdar *et al.* [22] proposed a distributed trust computing framework (DTCF) for vehicular ad-hoc networks (VANETs) to calculate the trust of each vehicle solely by direct observations of neighbors in order to isolate malevolent nodes. DTCF doesn't consider second-hand information; rather it considers message authentication verification as a direct trust metric. A tier-based broadcasting mechanism is implemented in order to propagate control messages and information about happening events. Simulation results show the efficiency of DTCF to detect message modification attacks with lower detection latency. Thorat and Kulkarni [23] addressed the issue of packet dropping attack with an uncertainty analysis framework (UAF) extended from ad-hoc on-demand distance vector (AODV) routing protocol. UAF uses direct and indirect observations in order to calculate network belief, disbelief and uncertainty (BDU) metrics based on probability of packet forwarding for neighbor nodes. Simulation results show the effect of mobility models, network density and selfish nodes on BDU values. The results indicate that as the network converges, UAF provides better knowledge about performance and security of the network by providing information about benign and distrusted nodes. In order to optimize memory resources and minimize communication overhead, Sargunavathi and Manickam [24] proposed a Collaborative Trust based Secure Routing (CTSR) protocol based on direct and indirect observations. CTSR adopts Dempster-Shafer approach for deriving trust values along with a two-way acknowledgement model. Information about suspicious nodes is disseminated to all the associated neighbor nodes by broadcasting a message. The authors also take into account such fabricated broadcast message propagated by malicious nodes. Simulation results depict the effectiveness of the proposed scheme under different network parameters. To discover multiple trusted paths, Li *et al.* [25] proposed an ad hoc on-demand trusted-path distance vector (AOTDV) protocol which is extended from AODV and ad hoc on-demand multipath distance vector (AOMDV) protocols. AOTDV provides two dimensional evaluation of paths based on hop counts and route trust values. Trust values are calculated from direct observations based on control packet forwarding ratio and data packet forwarding ratio. Simulation results show the effectiveness of AOTDV against blackhole, grayhole and modifications attacks. Jawhar *et al.* [26] extended dynamic source routing (DSR) protocol to devise a trust-based routing protocol for ad-hoc and sensor networks (TRAS). A route with highest trust factor is selected for data communication, while other back-up routes are used after failure of the preliminary route. The trust factor of a node is increased when it actively participates in the packet forwarding process. At the same time, the trust factor is also improved when a node participating in the data transmission receives positive acknowledgements from the destination. Simulation results depict that out of the two evaluated versions of TRAS viz. TRAS-25 and TRAS-50, TRAS-50 performs better in different scenarios. Kerrache *et al.* [27] proposed a hybrid trust-based framework for reliable data delivery and DoS defense (TFDD) for VANETs. TFDD takes into account trust weight and link stability to discover most trusted and nearest neighbor to forward data packets. Verification of neighbors' behavior is carried out by observation of the exchanged messages in order to improve trust relationship among nodes. Simulation results show that TFDD provides a high detection rate of dishonest nodes by meeting delay restrictions of VANET communications, even when high percentage of attackers are present in the network. A lightweight trust-based QoS routing algorithm (TQR) extended from AODV was devised by Wang *et al.* [28] in order to mitigate sequence number attacks. TQR aggregates link delay (includes transmission delay, propagation delay and waiting delay of a buffered packet) and trust value (direct and indirect observations for packet forwarding behavior) to establish a refined routing cost metric to find a quality trusted path during the route discovery process. Thus, TQR ensures the forwarding of the packets via trusted and least link delayed routes in order to support real time audio/video transmissions in ad-hoc networks. Simulation results show the improved performance of TQR in most scenarios against an existing scheme in terms of packet delivery ratio, average end-to-end delay, routing overhead and detection ratio. Li and Song [29] proposed an attack-resistant trust management scheme (ART) to secure VANETs against accidents and to support intelligent transportation applications. ART uses two distinct metrics for trust evaluations: data trust and node trust. Data trust is evaluated to assess trustworthiness of the traffic data aggregated from sensed data and collected data from multiple vehicles. Node trust is used to assess trustworthiness of a node to infer the capability of the node to fulfill its functionality and to figure out trustworthiness of the node's recommendations for other nodes. Simulation results demonstrate the accuracy and effectiveness of ART and its capability to cope with packet dropping attacks, bad mouthing attacks and zigzag attacks. Sethuraman and Kannan [30] proposed a refined trust energy-ad hoc on demand distance vector (ReTE-AODV) scheme which opts reliable and trusted path for sending data packets which consumes lower energy. The scheme computes direct and indirect trust values along with computation of energy value of nodes. The trust model incorporates Bayesian probability in order to handle ambiguity for acquiring the refined trust value. This energy efficient scheme promises to perform better against three existing schemes in terms of packet delivery ratio and end-to-end delay.

In spite of the presence of several schemes in the literature, there doesn't exist any scheme (to the best of our knowledge) which identifies patterns followed by adversaries (if any) during route discovery process in order to detect adversaries before they actually launch packet dropping attacks. Therefore, we proposed a scheme in [1] which incorporates a pattern discovery mechanism with a trust model to identify the attack patterns followed by adversaries in order to improve quality-of-service (QoS) in MANETs in hostile environments. In addition, the security during analysis is necessary [31]. In future, we may consider combining watermarking [32] and access control management [33] with ad-hoc routing protocols in order to strengthen the security along with enhancing network capacity [34].

III. WORKING OF TRS-PD

TRS-PD [1] is extended from AODV protocol which adopts a two tier approach in which an attack pattern discovery mechanism sits on the top of the trust model. In this section, we summarize the pattern discovery mechanism, the trust model and the routing process adopted by TRS-PD.

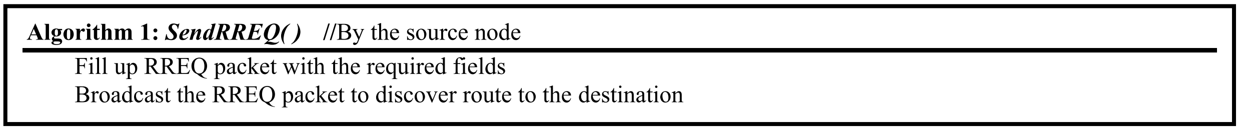
A. ATTACK-PATTERN DISCOVERY MECHANISM

In order to maximize the trust value in neighbor nodes, smart adversaries may pretend to be benign nodes by forwarding packets prior to launching packet dropping attacks. However, sequence number attacks launched by packet dropping adversaries may generate specific kind of patterns in fabricating some field values (such as sequence number and hop count)in the control packets during route discovery process. TRS-PD adopts a pattern discovery mechanism [1] to analyze the recorded field values from the overheard/received control packets. The field values are recorded in two sliding windows:(i) the first sliding window (SL1) records destination node's identity, current time, hop count and destination sequence number (ii) the second sliding window (SL2) records destination node's identity, current time and difference between the destination sequence numbers of the received reply packet and that of the corresponding request packet. An algorithm adopting the model of Method of Common Differences analyzes the recorded data and outputs whether the neighbor node follows any attack-pattern or not. Route discovery process is strengthened by this mechanism in isolating the blacklisted adversaries who may launch packet dropping attacks later on. However, it should be noted that malicious nodes may continue to drop packets until the monitoring node fills all slots of its sliding windows and predicts them as blacklisted nodes.

B. TRUST MODEL

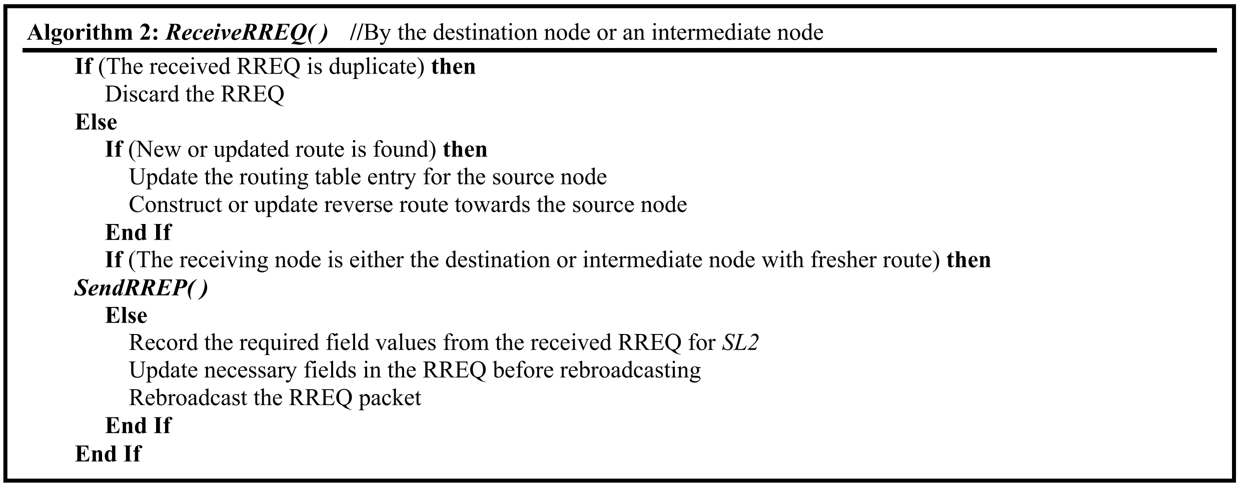
The second tier of the scheme is the trust model [1] which sits below the attack-pattern discovery mechanism. While adversaries following certain attack-patterns may get detected by the pattern discovery mechanism, the other packet dropping adversaries (which do not follow any pattern) are identified by the trust model.

TRS-PD uses direct as well as indirect trust in its trust framework. A node's historical distrust value is calculated by aggregating packet dropping ratios of control and data packets. If a monitoring node observes a neighboring node crossing the distrust threshold (η), it is temporarily marked



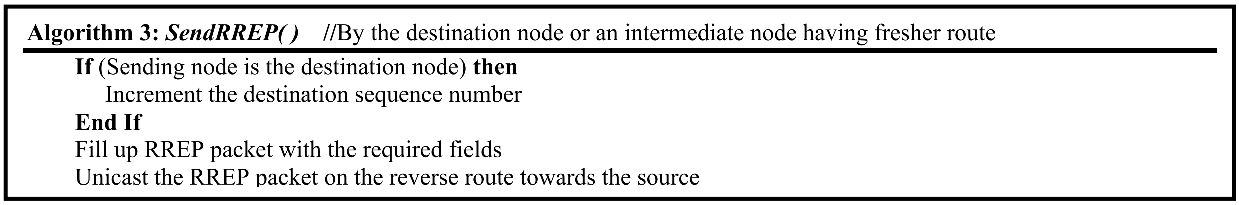
```
Algorithm 1: SendRREQ( )   //By the source node
    Fill up RREQ packet with the required fields
    Broadcast the RREQ packet to discover route to the destination
```

FIGURE 1. ***SendRREQ*** **procedure [1].**



```
Algorithm 2: ReceiveRREQ( )   //By the destination node or an intermediate node
    If (The received RREQ is duplicate) then
        Discard the RREQ
    Else
        If (New or updated route is found) then
            Update the routing table entry for the source node
            Construct or update reverse route towards the source node
        End If
        If (The receiving node is either the destination or intermediate node with fresher route) then
    SendRREP( )
        Else
            Record the required field values from the received RREQ for SL2
            Update necessary fields in the RREQ before rebroadcasting
            Rebroadcast the RREQ packet
        End If
    End If
```

FIGURE 2. ***ReceiveRREQ*** **procedure [1].**



```
Algorithm 3: SendRREP( )   //By the destination node or an intermediate node having fresher route
    If (Sending node is the destination node) then
        Increment the destination sequence number
    End If
    Fill up RREP packet with the required fields
    Unicast the RREP packet on the reverse route towards the source
```

FIGURE 3. ***SendRREP*** **procedure [1].**

as a blacklisted node. However, such a node regaining its trust is removed from the blacklist later on if it does not follow attack-patterns and trusted neighbors have also not recommended it as a distrusted node. TRS-PD distributes responsibility to individual nodes to construct a trusted forward route rather than imposing burden on the source node.

It is to be noted that in the real world, threshold value should be determined on the basis of the safety requirements of the application.

C. ROUTING PROCESS

TRS-PD modifies the routing process of AODV: (i) Fig. 1 [1] represents send request procedure (*sendRREQ*) (ii) Fig. 2 [1] represents receive request procedure (*receiveRREQ*) (iii) Fig. 3 [1] represents send reply procedure (*sendRREP*) (iv) Fig. 4 [1] represents receive reply procedure (*receiveRREP*).

D. PROCEDURES OF TRUST RECOMMENDATION AND TRUST UPDATE

The trust update procedure incorporates the attack-pattern discovery mechanism to verify the attack patterns along with the calculation of distrust values (DTV) of the neighbors. In the case of finding a distrusted next hop, the monitoring node initiates a route hand-off mechanism through local route discovery process to discover an alternate trusted route to the destination. Fig. 5 [1] represents the trust update procedure (*UpdateTrust*) of TRS-PD.

TRS-PD uses the HELLO messages in order to supply trust recommendations to the neighbor nodes. The supplied recommendations containing a list of blacklisted nodes are received from the HELLO message sent by the neighbor. If the supplier node is a trusted node, the recommendations are considered. Fig. 6 [1] represents the trust recommendation procedure (*RecommendTrust*) of TRS-PD.

IV. SIMULATION RESULTS AND ANALYSIS

The NS-2 network simulator is used to analyze the performance of TRS-PD by varying values of *distrust threshold*, *trust update interval* and *trust components' weights* (denoted as *w1* and *w2* for *control packet dropping ratio* and *data packet dropping ratio* respectively) under different network conditions. This sensitivity analysis of TRS-PD is carried out in the presence of malevolent nodes adopting three

```
Algorithm 4: ReceiveRREP( )   //By an intermediate node on the reverse route or the source node
    Record the required field values from the received RREP (or from the overheard RREP)
    Insert the corresponding recorded values into SL1 and SL2
    If (The neighbor sending RREP is marked as blacklisted) then
    Discard the RREP
    Else
      If (New or updated route is found) then
    Update the routing table entry for the destination node
    End If
      If (Receiving node is the source node) then
    Discard the RREP
     Send data through the forward route if the route is fresher and the next hop is trusted
      Else
        Forward the RREP packet on the reverse route towards the source
      End If
    End If
```

FIGURE 4. ***ReceiveRREP*** **procedure [1].**

```
Algorithm 5: UpdateTrust( ) //By each node
    For (Each neighbor table entry)
    Do
      Verify the existence of attack patterns from SL1 and SL2 of the neighbor
      Calculate DTV value of the neighbor
    If (The neighbor follows attack patterns or has $DTV>\eta$) then
           Mark the node as a distrusted node
    ElseIf (The neighbor doesn't follow attack pattern, has $DTV \leq \eta$ and recommended as trusted node) then
    Mark the node as a trusted node   // the node which had $DTV>\eta$ in the past, but found trusted now
         End If
    Done
    For (Each routing table entry)
    Do
      Find the entry of the next hop from the neighbor table
    If (The next hop is found to be distrusted in the neighbor table) then
      Discard the route containing the malevolent next hop
           Initiate a local route discovery process to discover an alternate route to the destination
    End If
    Done
```

FIGURE 5. ***UpdateTrust*** **procedure [1].**

distinct adversary models viz. *Attack1*, *Attack2* and *Attack3* [1]. *Attack1* operates in promiscuous mode and generates attack-patterns by fabricating 'hop count' information while sending a route reply packet during route discovery process. Meanwhile, *Attack2* generates attack-patterns by fabricating 'destination sequence number' while sending a route reply packet. On the other side, *Attack3* carries out random behavior and doesn't generate any attack-pattern during route discovery process. The operations performed by all three adversary models are discussed in [1] with pseudo-code.

In an area of 1500 m × 1500 m, benign nodes were randomly distributed which execute TRS-PD or AODV protocol. Randomly located attacker nodes selectively perform packet dropping attack by either adopting the first adversary model (*Attack1*), the second adversary model (*Attack2*) or the third adversary model (*Attack3*). We use User Datagram Protocol (UDP) as the transport protocol and Constant-Bit-Rate (CBR) as the traffic sources where the source nodes send4 packets per second. The experimental data represent an average value resulting from 10 distinct simulations. The major simulation parameters are represented in Table 1.

TABLE 1. **Simulation parameters.**

Parameter	Value
Coverage area	1500 m × 1500 m
MAC layer protocol	IEEE 802.11
Communication range of each node	250 m
Channel bandwidth	2 Mbps
Traffic type	CBR-UDP
Mobility model	Random way point
Number of nodes	100
Number of connections	10
Pause time	20 sec
Packet rates	4 packets/sec
Maximum mobility (varying)	5 m/sec ~ 25 m/sec
Percentage of malicious nodes (varying)	0% ~ 50%
Packet size(varying)	512~1024 bytes
Simulation time(varying)	200~ 1000 sec
Routing protocols	TRS-PD, AODV, Attack1, Attack2, Attack3

In order to find the correct tuning, we vary the values of *distrust threshold* ($\eta = 0.4$ and $\eta = 0.5$), *trust components'*

```
Algorithm 6: RecommendTrust( ) //By each node
//Before broadcasting a HELLO packet//
Construct an empty Blacklist for recommendation purpose
For (Each neighbor table entry)
Do
If (The neighbor is marked as a distrusted node) then
Insert the neighbor identity into the Blacklist
End If
Done
Incorporate the Blacklist into the HELLO packet
Broadcast the HELLO packet to the neighbors
//After receiving a HELLO packet//
Receive HELLO packet from the neighbor
If (The neighbor sending the HELLO packet is trusted) then
  Obtain the Blacklist from the HELLO packet
  For (Each entry in the Blacklist)
  Do
Find the corresponding entry in the neighbor table
    If (The neighbor entry exists) then
Set Recommendation value as 'distrusted' for the neighbor
End If
  Done
End If
```

FIGURE 6. ***RecommendTrust*** **procedure [1].**

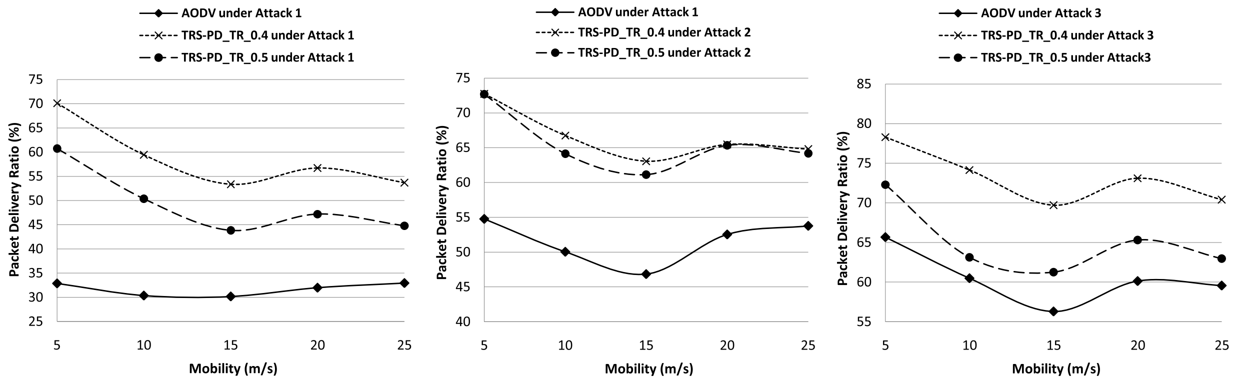


FIGURE 7. **PDR vs mobility with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.**

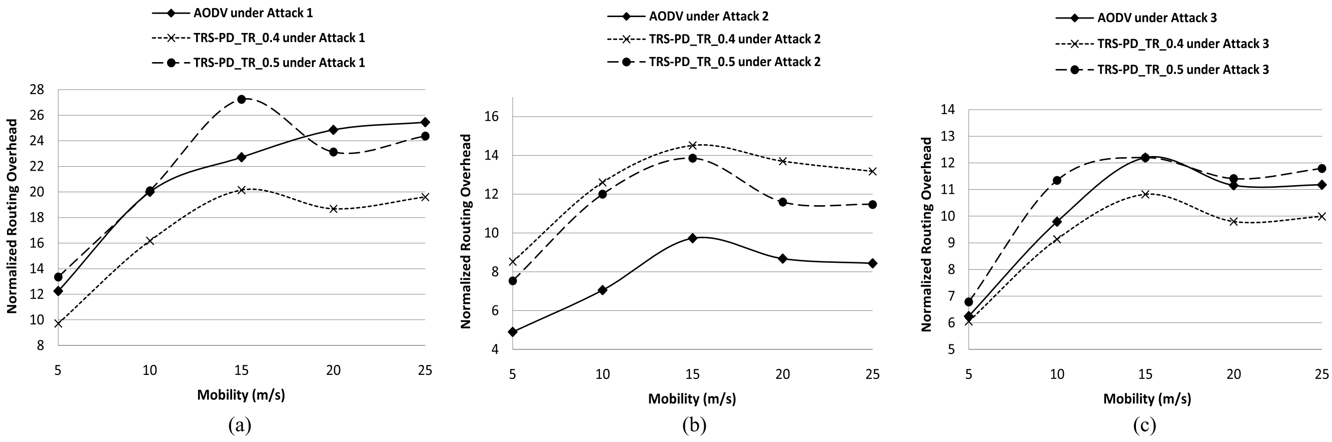
weights ($w1 = 0.5$, $w2 = 0.5$ and $w1 = 0.3$, $w2 = 0.7$) and *trust update interval* ($UI = 1$ sec and $UI = 3$ sec). For evaluating the performance of TRS-PD, the following metrics are used: (i) *Packet Delivery Ratio (PDR)* (ii) *Normalized Routing Overhead (NRO)*. In order to test TRS-PD under various network scenarios, we vary the following network parameters: (i) *Node Mobility* (ii) *Percentage of attackers* (iii) *Packet Size*(iv) *Simulation time*.

A. TEST1: VARYING NODE MOBILITY

In this test, we evaluate the performance of TRS-PD and AODV under *Attack1*, *Attack2* and *Attack3* by varying maximum speeds of nodes from 5 m/s to 25 m/s and keeping other simulation parameters fixed: number of malevolent nodes 20%, simulation time 200 sec and packet size 512 bytes.

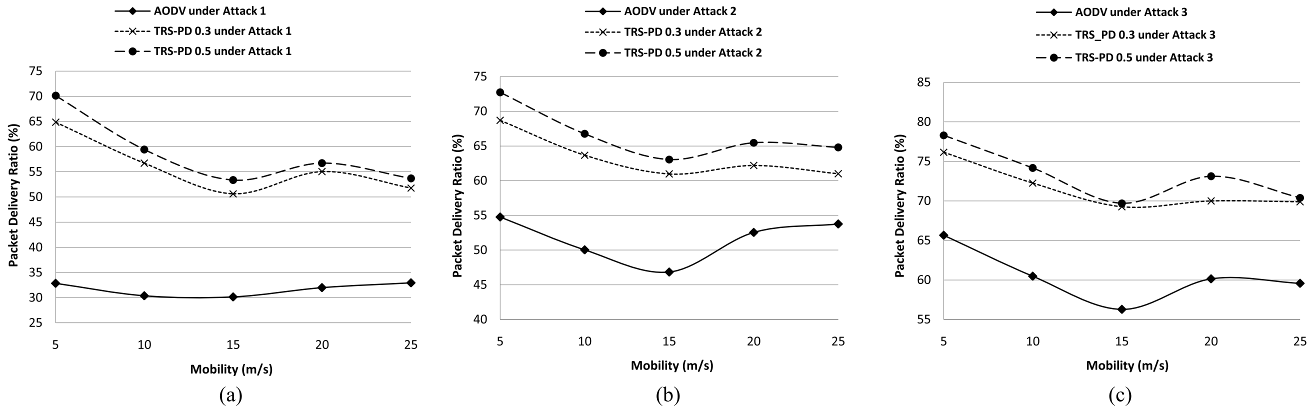
Fig. 7 depicts the PDR of TRS-PD under the three adversary models with increasing mobility by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 7 (a) shows that, under *Attack1*, PDR of AODV varies in the range of 30.15 and 32.95, while PDR of TRS-PD with $\eta = 0.4$ decreases from 70.13 to 53.35 and PDR of TRS-PD with $\eta = 0.5$ decreases from 60.74 to 43.82. Fig. 7 (b) shows that, under *Attack2*, PDR of AODV varies in the range of 54.77 and 50.03, while PDR of TRS-PD with $\eta = 0.4$ decreases from 72.72 to 63.06 and PDR of TRS-PD with $\eta = 0.5$ decreases from 72.68 to 61.13. Fig. 7 (c) shows that, under *Attack3*, PDR of AODV varies in the range of 65.66 and 56.28, while PDR of TRS-PD with $\eta = 0.4$ decreases from 78.31 to 69.71 and PDR of TRS-PD with $\eta = 0.5$ decreases from 72.31 to 61.25. We can analyze that PDR of TRS-PD under distinct adversary models is different as mode of operations adopted by them is distinct. Meanwhile, under all three adversaries, the average PDR provided by TRS-PD with $\eta = 0.4$ is significantly better than TRS-PD with $\eta = 0.5$.

Fig. 8 depicts the NRO of TRS-PD under the three adversary models with increasing mobility by taking $\eta = 0.4$ and



(a) (b) (c)

FIGURE 8. NRO vs mobility with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.



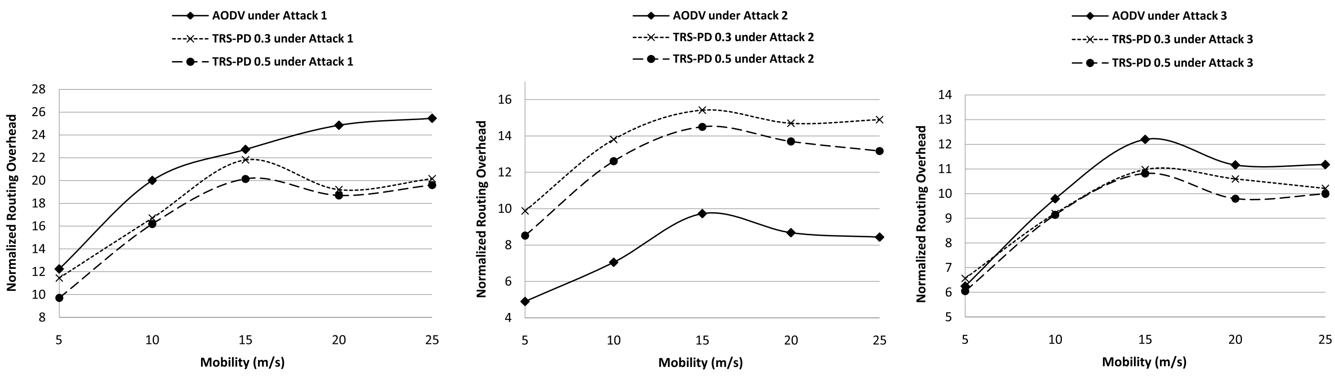
(a) (b) (c)

FIGURE 9. PDR vs mobility with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

$\eta = 0.5$. Fig. 8 (a) shows that, under *Attack1*, NRO of AODV increases from 12.25 to 25.45, while NRO of TRS-PD with $\eta = 0.4$ increases from 9.71 to 20.15 and NRO of TRS-PD with $\eta = 0.5$ increases from 13.35 to 27.24. Fig. 8 (b) shows that, under *Attack2*, NRO of AODV increases from 4.90 to 9.73, while NRO of TRS-PD with $\eta = 0.4$ increases from 8.52 to 14.51 and NRO of TRS-PD with $\eta = 0.5$ increases from 7.54 to 13.85. Fig. 8 (c) shows that, under *Attack3*, NRO of AODV increases from 6.25 to 12.20, while NRO of TRS-PD with $\eta = 0.4$ increases from 6.05 to 10.82 and NRO of TRS-PD with $\eta = 0.5$ increases from 6.78 to 12.20. We can analyze that NRO of TRS-PD under distinct adversary models is different as number of control packets generated under distinct adversary models are different. Meanwhile, under *Attack1* and *Attack3*, TRS-PD with $\eta = 0.4$ provides significant improvement in NRO as compared to TRS-PD with $\eta = 0.5$, while under *Attack2*, TRS-PD with $\eta = 0.5$ performs marginally better.

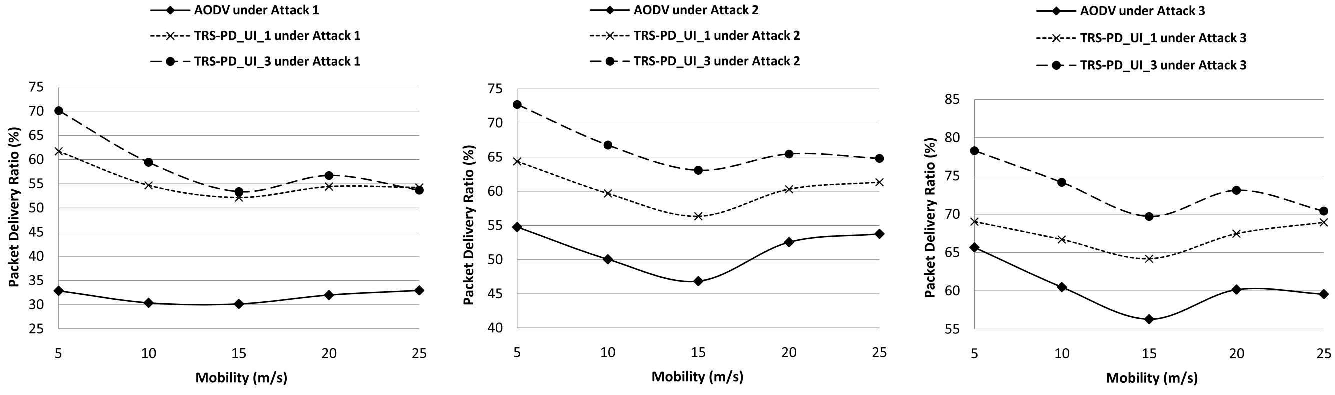
Fig. 9 depicts the PDR of TRS-PD under the three adversary models with increasing mobility by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 9(a) shows that, under *Attack1*, PDR of TRS-PD with $w1 = 0.3$ decreases from 64.85 to 50.64 and PDR of TRS-PD with $w1 = 0.5$ decreases from 70.13 to 53.35. Fig.9 (b) shows that, under *Attack2*, PDR of TRS-PD with $w1 = 0.3$ decreases from 68.69 to 60.97 and PDR of TRS-PD with $w1 = 0.5$ decreases from 72.72 to 63.06. Fig.9 (c) shows that, under *Attack3*, PDR of TRS-PD with $w1 = 0.3$ decreases from 76.17 to 69.27 and PDR of TRS-PD with $w1 = 0.5$ decreases from 78.31 to 69.71. We can analyzethat, under all three adversaries, the average PDR provided by TRS-PD with $w1 = 0.5$ is significantly better than TRS-PD with $w1 = 0.3$.

Fig. 10 depicts the NRO of TRS-PD under the three adversary models with increasing mobility by taking $w1 = 0.3$ and $w1 = 0.5$. Fig.10 (a) shows that, under *Attack1*, NRO of TRS-PD with $w1 = 0.3$ increases from 11.46 to 21.81 and NRO of TRS-PD with $w1 = 0.5$ increases from 9.71 to 20.15. Fig.10 (b) shows that, under *Attack2*, NRO of TRS-PD with $w1 = 0.3$ increases from 9.88 to 15.42 and NRO of TRS-PD with $w1 = 0.5$ increases from 8.52 to 14.51. Fig.10(c) shows that, under *Attack3*, NRO of TRS-PD with $w1 = 0.3$ increases from 6.57 to 10.98 and NRO of TRS-PD with $w1 = 0.5$ increases from 6.05 to 10.82. We can analyze that TRS-PD with $w1 = 0.5$ provides significant improvement in



(a) (b) (c)

FIGURE 10. NRO vs mobility with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.



(a) (b) (c)

FIGURE 11. PDR vs mobility with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

NRO as compared to TRS-PD with $w1 = 0.3$ under all three adversary models.

Fig. 11 depicts the PDR of TRS-PD under the three adversary models with increasing mobility by taking $UI = 1$ and $UI = 3$. Fig.11 (a) shows that, under *Attack1*, PDR of TRS-PD with $UI = 1$ decreases from 61.69 to 52.16 and PDR of TRS-PD with $UI = 3$ decreases from 70.13 to 53.35. Fig.11 (b) shows that, under *Attack2*, PDR of TRS-PD with $UI = 1$ decreases from 64.38 to 56.34 and PDR of TRS-PD with $UI = 3$ decreases from 72.72 to 63.06. Fig.11 (c) shows that, under *Attack3*, PDR of TRS-PD with $UI = 1$ decreases from 69.03 to 64.19 and PDR of TRS-PD with $UI = 3$ decreases from 78.31 to 69.71. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

Fig. 12 depicts the NRO of TRS-PD under the three adversary models with increasing mobility by taking $UI = 1$ and $UI = 3$. Fig.12 (a) shows that, under *Attack1*, NRO of TRS-PD with $UI = 1$ increases from 14.36 to 23.78 and NRO of TRS-PD with $UI = 3$ increases from 9.71 to 20.15. Fig.12 (b) shows that, under *Attack2*, NRO of TRS-PD with $UI = 1$ increases from 13.03 to 20.32 and NRO of TRS-PD with $UI = 3$ increases from 8.52 to 14.51. Fig.12 (c) shows that, under *Attack3*, NRO of TRS-PD with $UI = 1$ increases from 9.26 to 14.40 and NRO of TRS-PD with $UI = 3$ increases from 6.05 to 10.82. We can analyze that TRS-PD with $UI = 3$ provides significant improvement in NRO as compared to TRS-PD with $UI = 1$ under all three adversary models.

B. TEST 2: VARYING PERCENTAGE OF MALICIOUS NODES

In this test, we evaluate the performance of TRS-PD and AODV under *Attack1*, *Attack2* and *Attack3* by varying percentage of attackers from 10% to 50% and keeping other simulation parameters fixed: mobility 10 m/sec, simulation time 200 sec and packet size 512 bytes.

Fig. 13 depicts the PDR of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 13 (a) shows that, under *Attack1*, PDR of AODV decreases from 36.51 to 19.60, while PDR of TRS-PD with $\eta = 0.4$ decreases from 70.59 to 37.26 and PDR of TRS-PD with $\eta = 0.5$ decreases from 63.70 to 28.99. Fig. 13 (b) shows that, under *Attack2*, PDR of AODV decreases from 56.37 to 38.71, while PDR of

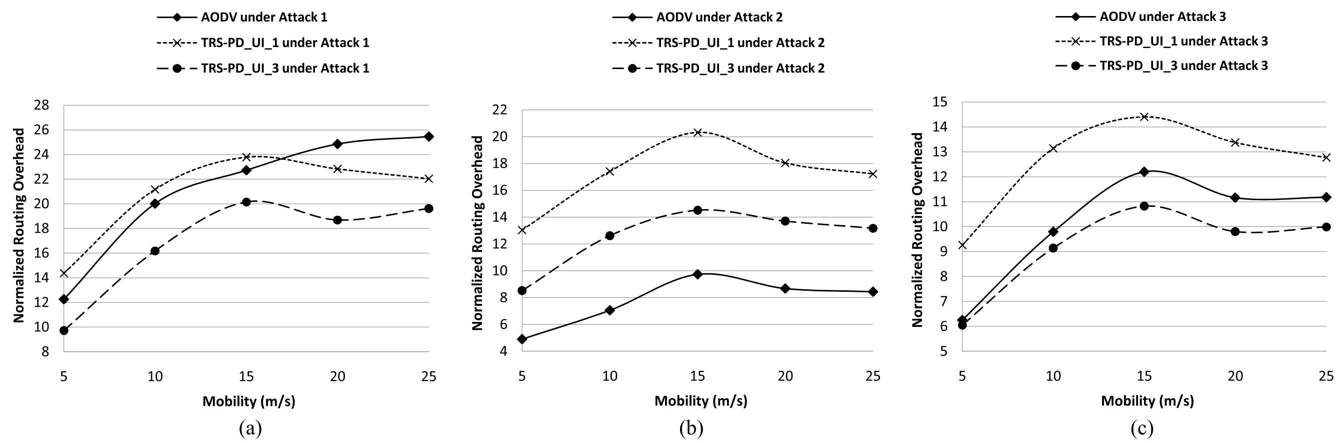


FIGURE 12. **NRO vs mobility with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.**

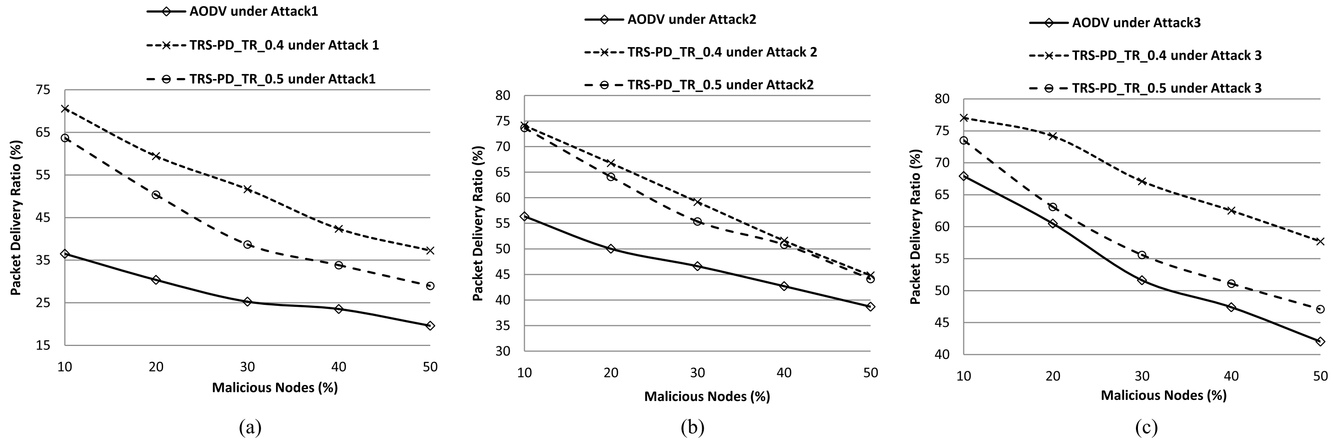


FIGURE 13. **PDR vs percentage of adversaries with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.**

TRS-PD with $\eta = 0.4$ decreases from 74.12 to 44.82 and PDR of TRS-PD with $\eta = 0.5$ decreases from 73.66 to 44.14. Fig. 13 (c) shows that, under *Attack3*, PDR of AODV decreases from 67.93 to 42.02, while PDR of TRS-PD with $\eta = 0.4$ decreases from 77.04 to 57.71 and PDR of TRS-PD with $\eta = 0.5$ decreases from 73.53 to 47.08. We can analyze that PDR of TRS-PD under distinct adversary models is different due to the aforementioned reason. Meanwhile, under all three adversaries, the average PDR provided by TRS-PD with $\eta = 0.4$ is significantly better than TRS-PD with $\eta = 0.5$.

Fig. 14 depicts the NRO of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $\eta = 0.4$ and $\eta = 0.5$. Fig.14 (a) shows that, under *Attack1*, NRO of AODV varies between 16.11 and 20.11, while NRO of TRS-PD with $\eta = 0.4$ increases from 12.51 to 21.44 and NRO of TRS-PD with $\eta = 0.5$ increases from 14.61 to 32.81. Fig.14 (b) shows that, under *Attack2*, NRO of AODV increases from 6.63 to 7.70, while NRO of TRS-PD with $\eta = 0.4$ increases from 11.22 to 17.26 and NRO of TRS-PD with $\eta = 0.5$ increases from 10.10 to 14.59. Fig.14 (c) shows that, under *Attack3*, NRO of AODV varies between 8.92 and10.72, while NRO of TRS-PD with $\eta = 0.4$ decreases from 9.43 to 6.20 and NRO of TRS-PD with $\eta = 0.5$ varies between 9.41 and 12.37. We can analyze that NRO of TRS-PD under distinct adversary models is different as per the aforementioned reasons. Meanwhile, under *Attack1* and *Attack3*, TRS-PD with $\eta = 0.4$ provides significant improvement in NRO as compared to TRS-PD with $\eta = 0.5$, while under *Attack2*, TRS-PD with $\eta = 0.5$ performs marginally better.

Fig. 15 depicts the PDR of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 15 (a) shows that, under *Attack1*, PDR of TRS-PD with $w1 = 0.3$ decreases from 66.98 to 36.69 and PDR of TRS-PD with $w1 = 0.5$ decreases from 70.59 to 37.26. Fig. 15 (b) shows that, under *Attack2*, PDR of TRS-PD with $w1 = 0.3$ decreases from 68.90 to 43.42 and PDR of TRS-PD with $w1 = 0.5$ decreases from 74.12 to 44.82. Fig. 15 (c) shows that, under *Attack3*, PDR of TRS-PD with $w1 = 0.3$ decreases from 76.29 to 55.97 and PDR of TRS-PD

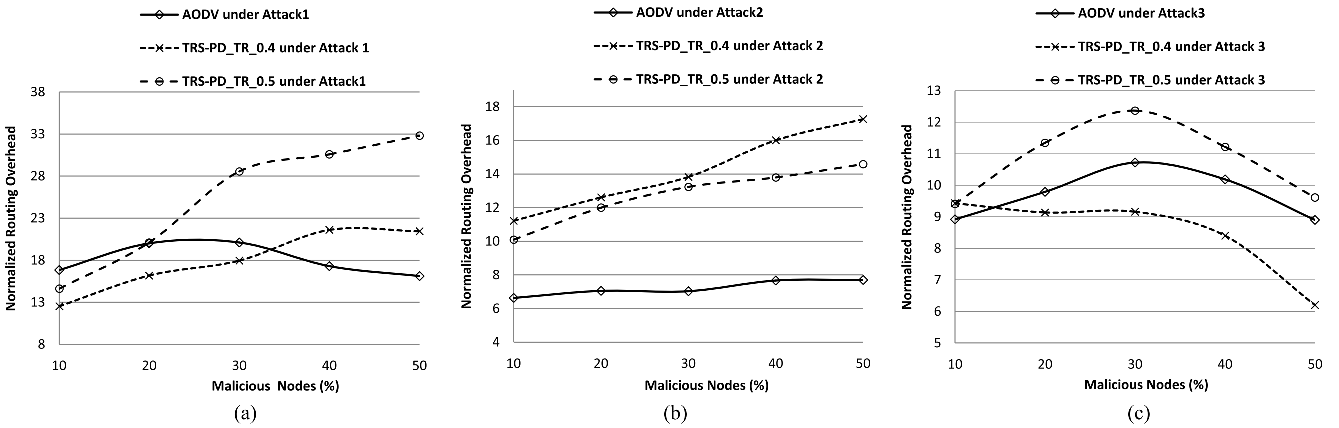


FIGURE 14. NRO vs percentage of adversaries with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

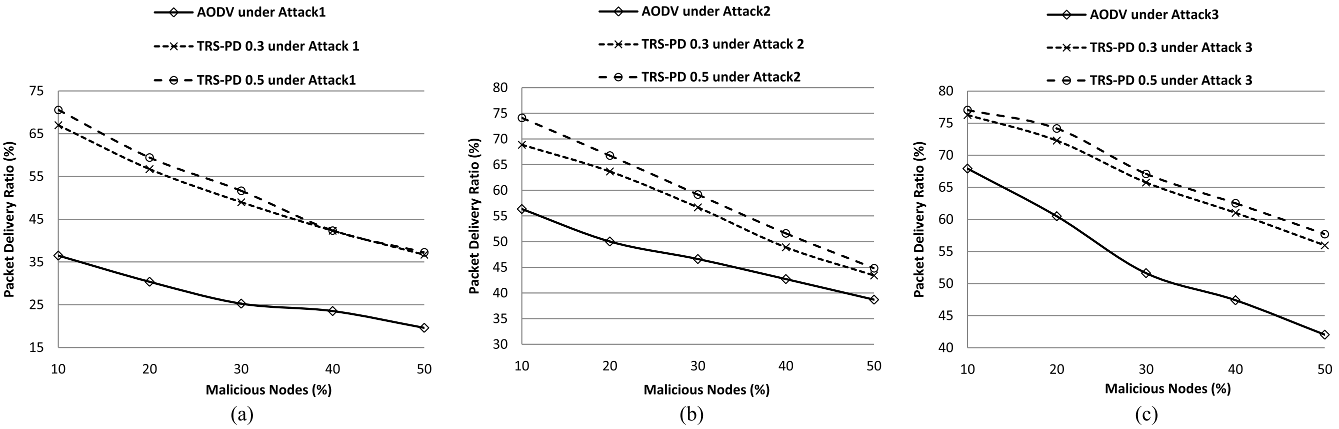


FIGURE 15. PDR vs percentage of adversaries with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

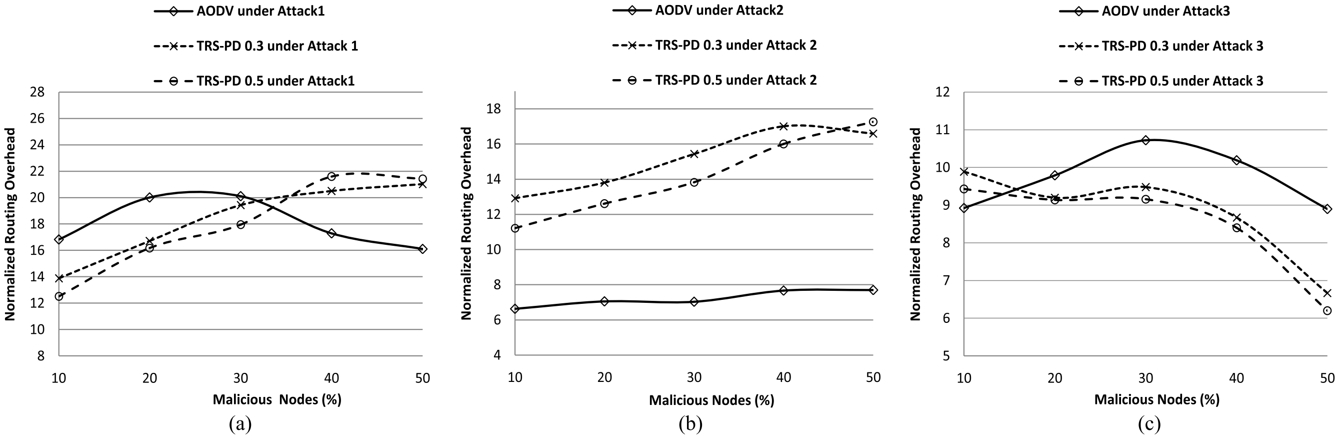
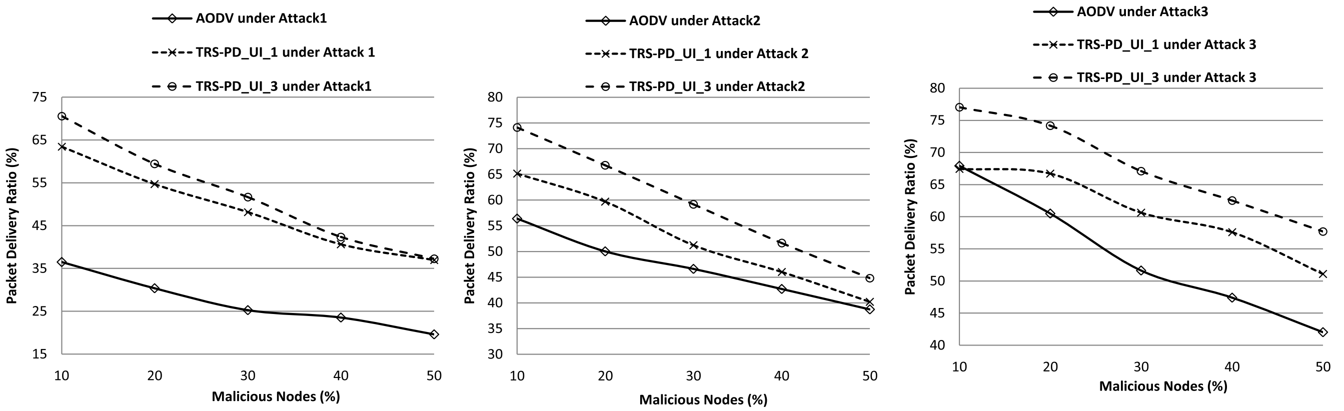


FIGURE 16. NRO vs percentage of adversaries with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

with $w1 = 0.5$ decreases from 77.04 to 57.71. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $w1 = 0.5$ is significantly better than TRS-PD with $w1 = 0.3$.

Fig. 16 depicts the NRO of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 16 (a) shows that, under *Attack1*, NRO of

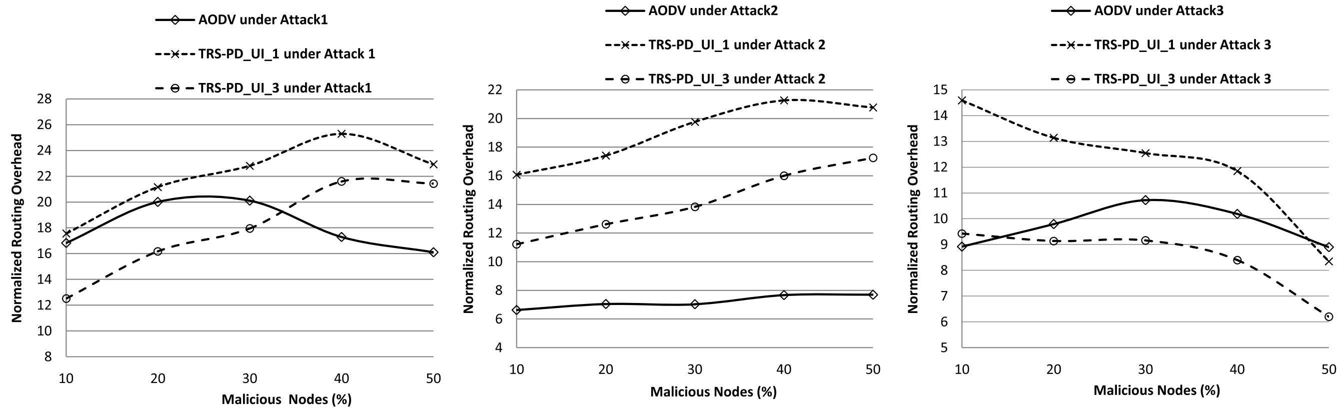


(a)

(b)

(c)

FIGURE 17. PDR vs percentage of adversaries with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.



(a)

(b)

(c)

FIGURE 18. NRO vs percentage of adversaries with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

TRS-PD with $w1 = 0.3$ increases from 13.88 to 21.02 and NRO of TRS-PD with $w1 = 0.5$ increases from 12.51 to 21.44. Fig. 16 (b) shows that, under *Attack2*, NRO of TRS-PD with $w1 = 0.3$ increases from 12.92 to 17.01 and NRO of TRS-PD with $w1 = 0.5$ increases from 11.22 to 17.26. Fig. 16 (c) shows that, under *Attack3*, NRO of TRS-PD with $w1 = 0.3$ decreases from 9.89 to 6.67 and NRO of TRS-PD with $w1 = 0.5$ decreases from 9.43 to 6.20. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $w1 = 0.5$ is considerably better than TRS-PD with $w1 = 0.3$.

Fig. 17 depicts the PDR of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $UI = 1$ and $UI = 3$. Fig. 17 (a) shows that, under *Attack1*, PDR of TRS-PD with $UI = 1$ decreases from 63.45 to 36.97 and PDR of TRS-PD with $UI = 3$ decreases from 70.59 to 37.26. Fig. 17 (b) shows that, under *Attack2*, PDR of TRS-PD with $UI = 1$ decreases from 65.15 to 40.20 and PDR of TRS-PD with $UI = 3$ decreases from 74.12 to 44.82. Fig. 17 (c) shows that, under *Attack3*, PDR of TRS-PD with $UI = 1$ decreases from 67.45 to 51.08 and PDR of TRS-PD with $UI = 3$ decreases from 77.04 to 57.71. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

Fig. 18 depicts the NRO of TRS-PD under the three adversary models with increasing percentage of adversaries by taking $UI = 1$ and $UI = 3$. Fig. 18 (a) shows that, under *Attack1*, NRO of TRS-PD with $UI = 1$ increases from 17.56 to 25.29 and NRO of TRS-PD with $UI = 3$ increases from 12.51 to 21.60. Fig. 18 (b) shows that, under *Attack2*, NRO of TRS-PD with $UI = 1$ increases from 16.08 to 21.26 and NRO of TRS-PD with $UI = 3$ increases from 11.22 to 17.26. Fig. 18 (c) shows that, under *Attack3*, NRO of TRS-PD with $UI = 1$ decreases from 14.59 to 8.35 and NRO of TRS-PD with $UI = 3$ decreases from 9.43 to 6.20. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

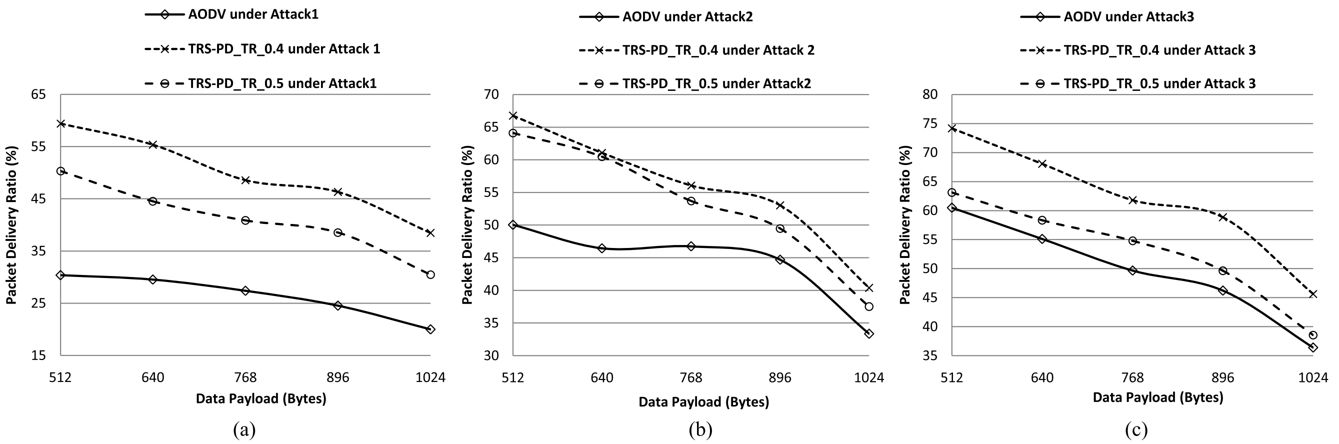


FIGURE 19. PDR vs data payload with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

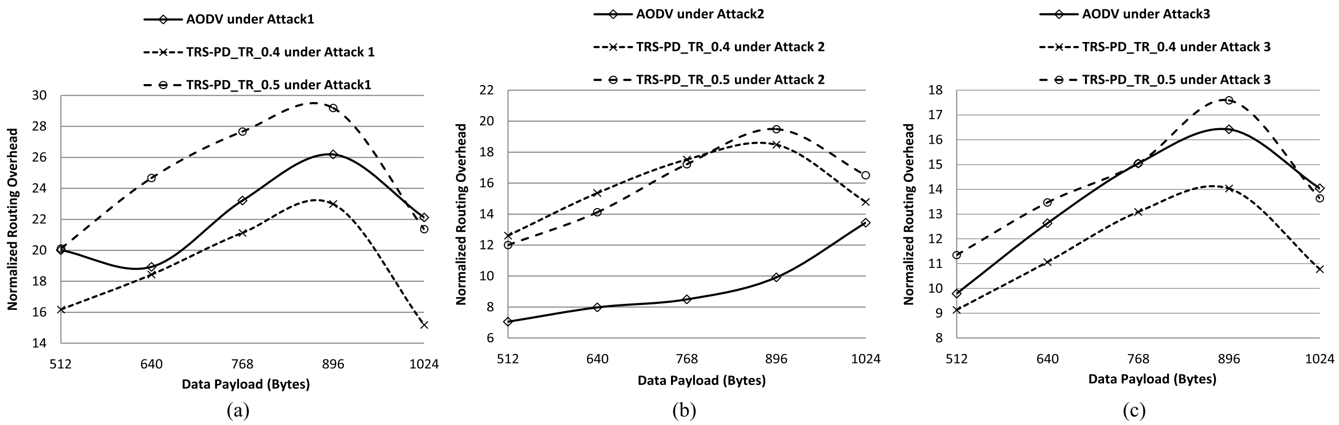


FIGURE 20. NRO vs data payload with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

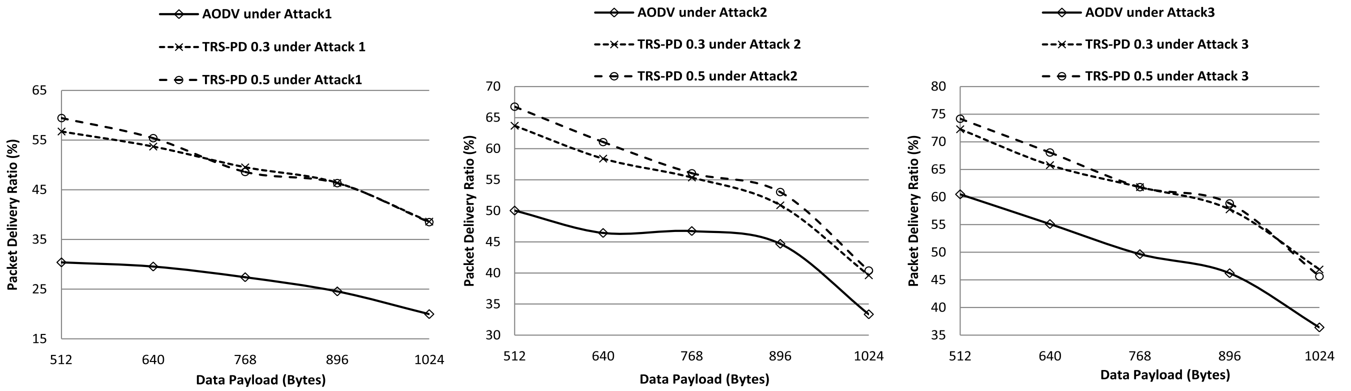
C. TEST 3: VARYING DATA PAYLOAD

In this test, we evaluate the performance of TRS-PD and AODV under *Attack1*, *Attack2* and *Attack3* by varying packet size from 512 bytes to 1024 bytes and keeping other simulation parameters fixed: mobility 10 m/sec, simulation time 200 sec and percentage of adversaries 20%.

Fig. 19 depicts the PDR of TRS-PD under the three adversary models with increasing payload by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 19 (a) shows that, under *Attack1*, PDR of AODV decreases from 30.37 to 19.98, while PDR of TRS-PD with $\eta = 0.4$ decreases from 59.43 to 38.47 and PDR of TRS-PD with $\eta = 0.5$ decreases from 50.35 to 30.49. Fig. 19 (b) shows that, under *Attack2*, PDR of AODV decreases from 50.03 to 33.36, while PDR of TRS-PD with $\eta = 0.4$ decreases from 66.76 to 40.39 and PDR of TRS-PD with $\eta = 0.5$ decreases from 64.12 to 37.51. Fig. 19 (c) shows that, under *Attack3*, PDR of AODV decreases from 60.48 to 36.38, while PDR of TRS-PD with $\eta = 0.4$ decreases from 74.16 to 45.65 and PDR of TRS-PD with $\eta = 0.5$ decreases from 63.12 to 38.56. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $\eta = 0.4$ is significantly better than TRS-PD with $\eta = 0.5$.

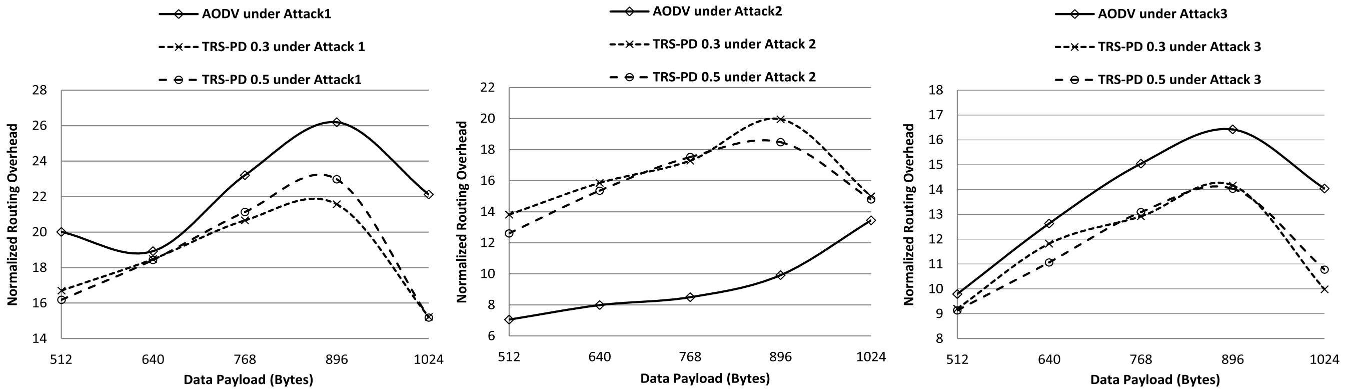
Fig. 20 depicts the NRO of TRS-PD under the three adversary models with increasing payload by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 20 (a) shows that, under *Attack1*, NRO of AODV varies between 18.93 and 26.19, while NRO of TRS-PD with $\eta = 0.4$ varies between 16.18 and 22.98, and NRO of TRS-PD with $\eta = 0.5$ varies between 20.11 and 29.18. Fig. 20 (b) shows that, under *Attack2*, NRO of AODV increases from 7.05 to 13.44, while NRO of TRS-PD with $\eta = 0.4$ increases from 12.61 to 18.48 and NRO of TRS-PD with $\eta = 0.5$ increases from 12 to 19.49. Fig. 20 (c) shows that, under *Attack3*, NRO of AODV increases from 9.79 to 16.42, while NRO of TRS-PD with $\eta = 0.4$ increases from 9.14 to 14.03 and NRO of TRS-PD with $\eta = 0.5$ increases from 11.35 to 17.59. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $\eta = 0.4$ is remarkably better than TRS-PD with $\eta = 0.5$.

Fig. 21 depicts the PDR of TRS-PD under the three adversary models with increasing payload by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 21 (a) shows that, under *Attack1*, PDR of TRS-PD with $w1 = 0.3$ decreases from 56.72 to 38.58 and PDR of TRS-PD with $w1 = 0.5$ decreases from 59.43 to 38.47.



(a) (b) (c)

FIGURE 21. PDR vs data payload with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.



(a) (b) (c)

FIGURE 22. NRO vs data payload with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

Fig. 21 (b) shows that, under *Attack2*, PDR of TRS-PD with $w1 = 0.3$ decreases from 63.68 to 39.62 and PDR of TRS-PD with $w1 = 0.5$ decreases from 66.76 to 40.39. Fig. 21 (c) shows that, under *Attack3*, PDR of TRS-PD with $w1 = 0.3$ decreases from 72.27 to 46.92 and PDR of TRS-PD with $w1 = 0.5$ decreases from 74.16 to 45.65. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $w1 = 0.5$ is marginally better than TRS-PD with $w1 = 0.3$.

Fig. 22 depicts the NRO of TRS-PD under the three adversary models with increasing payload by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 22 (a) shows that, under *Attack1*, NRO of TRS-PD with $w1 = 0.3$ varies between 16.71 and 21.57, and NRO of TRS-PD with $w1 = 0.5$ varies between 16.18 and 22.98. Fig. 22 (b) shows that, under *Attack2*, NRO of TRS-PD with $w1 = 0.3$ varies between 13.81 and 19.96, and NRO of TRS-PD with $w1 = 0.5$ varies between 12.61 and 18.48. Fig. 22 (c) shows that, under *Attack3*, NRO of TRS-PD with $w1 = 0.3$ varies between 9.20 and 14.17, and NRO of TRS-PD with $w1 = 0.5$ varies between 9.14 and 14.03. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $w1 = 0.5$ is slightly better than TRS-PD with $w1 = 0.3$.

Fig. 23 depicts the PDR of TRS-PD under the three adversary models with increasing payload by taking $UI = 1$ and $UI = 3$. Fig. 23 (a) shows that, under *Attack1*, PDR of TRS-PD with $UI = 1$ decreases from 54.68 to 35.93 and PDR of TRS-PD with $UI = 3$ decreases from 59.43 to 38.47. Fig. 23 (b) shows that, under *Attack2*, PDR of TRS-PD with $UI = 1$ decreases from 59.68 to 36.94 and PDR of TRS-PD with $UI = 3$ decreases from 66.76 to 40.39. Fig. 23 (c) shows that, under *Attack3*, PDR of TRS-PD with $UI = 1$ decreases from 66.71 to 42.99 and PDR of TRS-PD with $UI = 3$ decreases from 74.16 to 45.65. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

Fig. 24 depicts the NRO of TRS-PD under the three adversary models with increasing payload by taking $UI = 1$ and $UI = 3$. Fig. 24 (a) shows that, under *Attack1*, NRO of TRS-PD with $UI = 1$ varies between 18.54 and 26.74 and NRO of TRS-PD with $UI = 3$ varies between 15.19 and 22.98. Fig. 24 (b) shows that, under *Attack2*, NRO of TRS-PD with $UI = 1$ varies between 16.98 and 24.35, and NRO

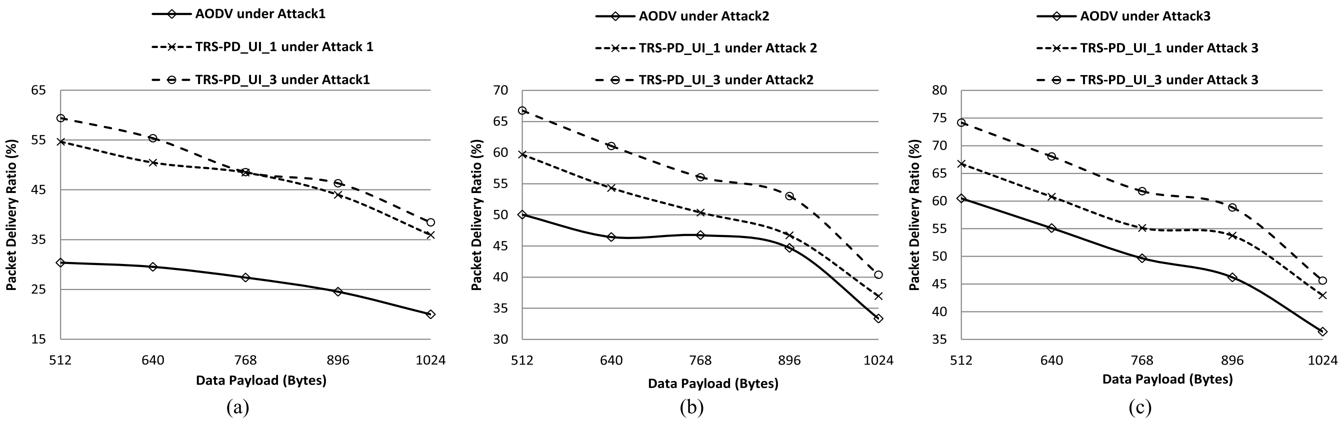


FIGURE 23. PDR vs data payload with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

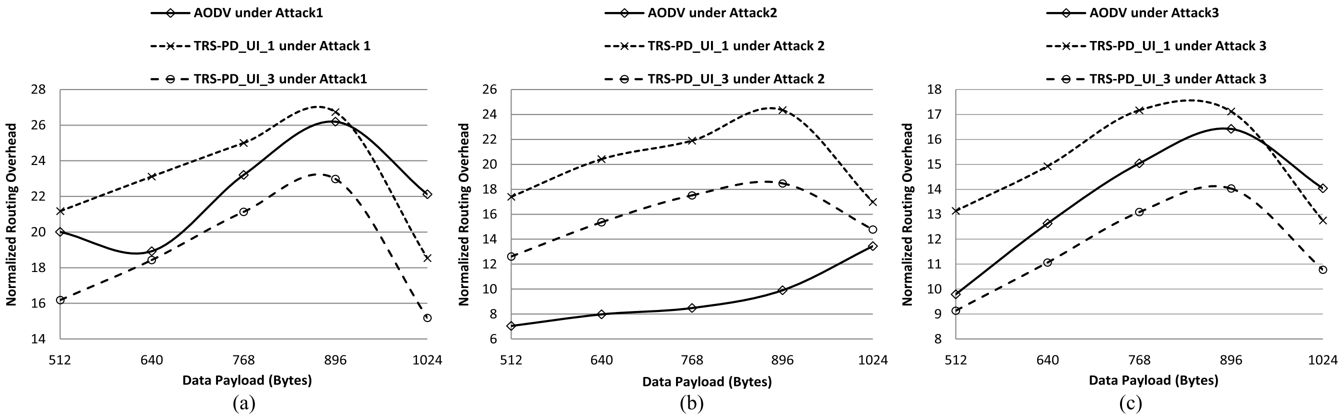


FIGURE 24. NRO vs data payload with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

of TRS-PD with $UI = 3$ varies between 12.61 and 18.48. Fig. 24 (c) shows that, under *Attack3*, NRO of TRS-PD with $UI = 1$ varies between 12.75 and 17.17, and NRO of TRS-PD with $UI = 3$ varies between 9.14 and 14.03. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

D. TEST 4: VARYING SIMULATION TIME

In this test, we evaluate the performance of TRS-PD and AODV under *Attack1*, *Attack2* and *Attack3* by varying simulation time from 200 sec to 1000 sec and keeping other simulation parameters fixed: mobility 10 m/sec, packet size 512 bytes and percentage of adversaries 20%.

Fig. 25 depicts the PDR of TRS-PD under the three adversary models with increasing simulation time by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 25 (a) shows that, under *Attack1*, PDR of AODV decreases from 30.37 to 25.17, while PDR of TRS-PD with $\eta = 0.4$ decreases from 59.43 to 54.40 and PDR of TRS-PD with $\eta = 0.5$ decreases from 50.35 to 47.96. Fig. 25 (b) shows that, under *Attack2*, PDR of AODV decreases from 50.03 to 45.44, while PDR of TRS-PD with $\eta = 0.4$ decreases from 66.76 to 61.36 and PDR of TRS-PD with $\eta = 0.5$ decreases from 64.12 to 60.69. Fig. 25 (c) shows that, under *Attack3*, PDR of AODV decreases from 60.48 to 56.98, while PDR of TRS-PD with $\eta = 0.4$ decreases from 74.16 to 71.13 and PDR of TRS-PD with $\eta = 0.5$ varies between66.62 and 63.12. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $\eta = 0.4$ is significantly higher than TRS-PD with $\eta = 0.5$.

Fig. 26 depicts the NRO of TRS-PD under the three adversary models with increasing simulation time by taking $\eta = 0.4$ and $\eta = 0.5$. Fig. 26 (a) shows that, under *Attack1*, NRO of AODV increases from 20.01 to 44.44, while NRO of TRS-PD with $\eta = 0.4$ increases from 16.18 to 20.53, and NRO of TRS-PD with $\eta = 0.5$ increases from 20.11 to 25.22. Fig. 26 (b) shows that, under *Attack2*, NRO of AODV increases from 7.05 to 23.73, while NRO of TRS-PD with $\eta = 0.4$ increases from 12.61 to 15.34 and NRO of TRS-PD with $\eta = 0.5$ increases from 12 to 15.94. Fig. 26 (c) shows that, under *Attack3*, NRO of AODV increases from 9.79 to 11.98, while NRO of TRS-PD with $\eta = 0.4$ varies between 8.49 and 9.50, and NRO of TRS-PD with $\eta = 0.5$ varies between 9.77 to 12.07. We can analyze that, under all three

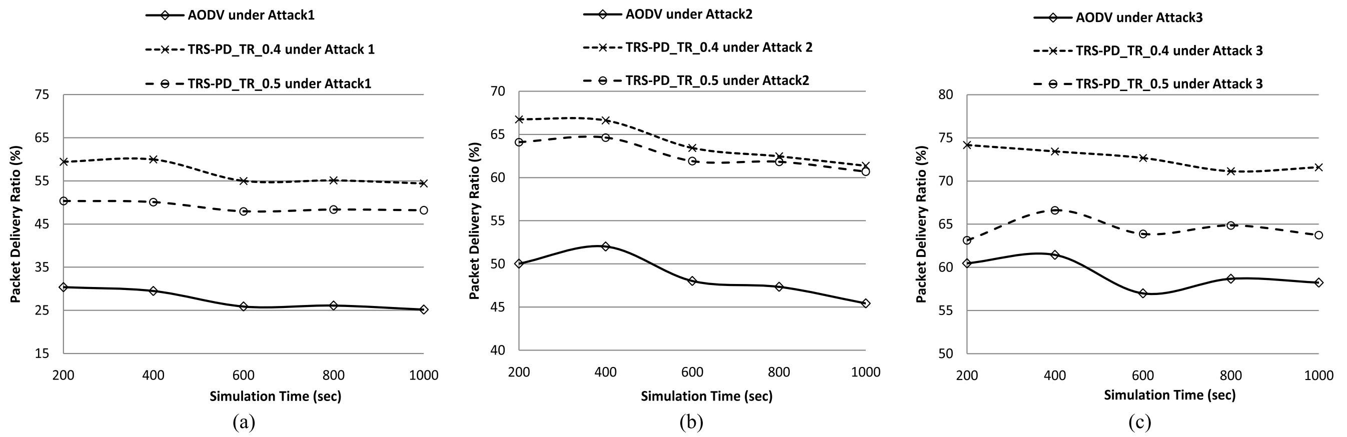


FIGURE 25. PDR vs simulation time with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

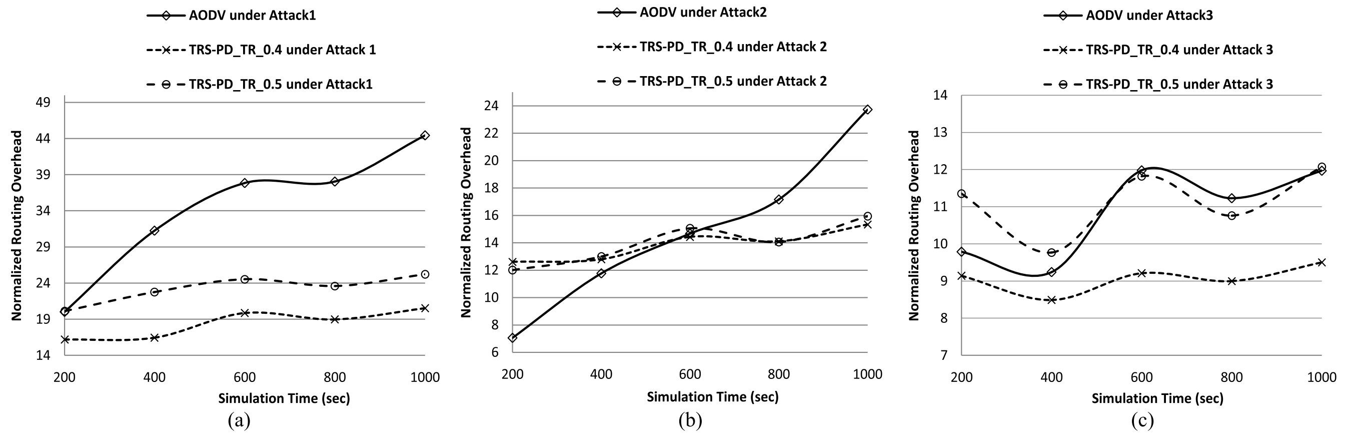


FIGURE 26. NRO vs simulation time with distinct threshold values. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

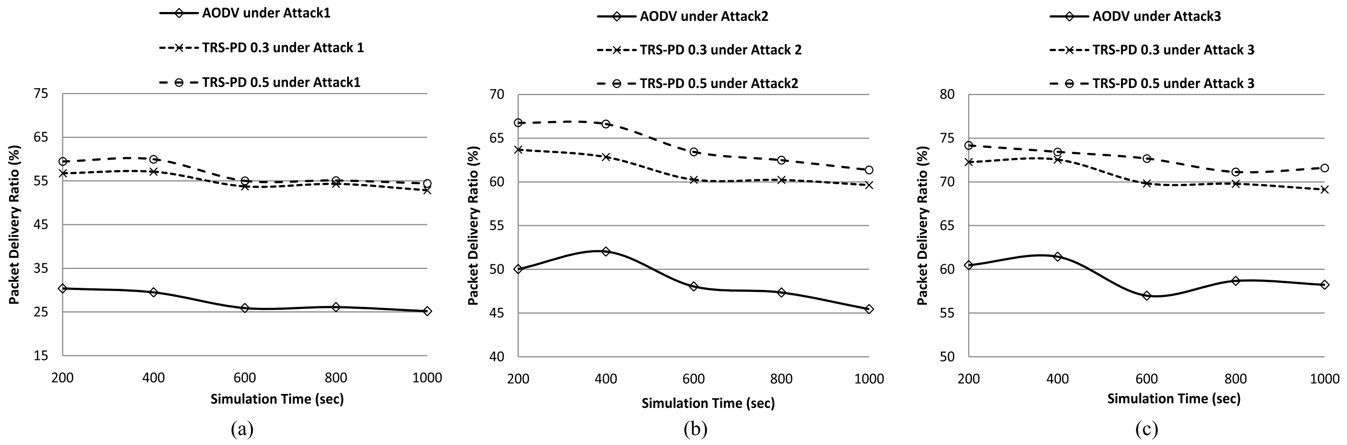
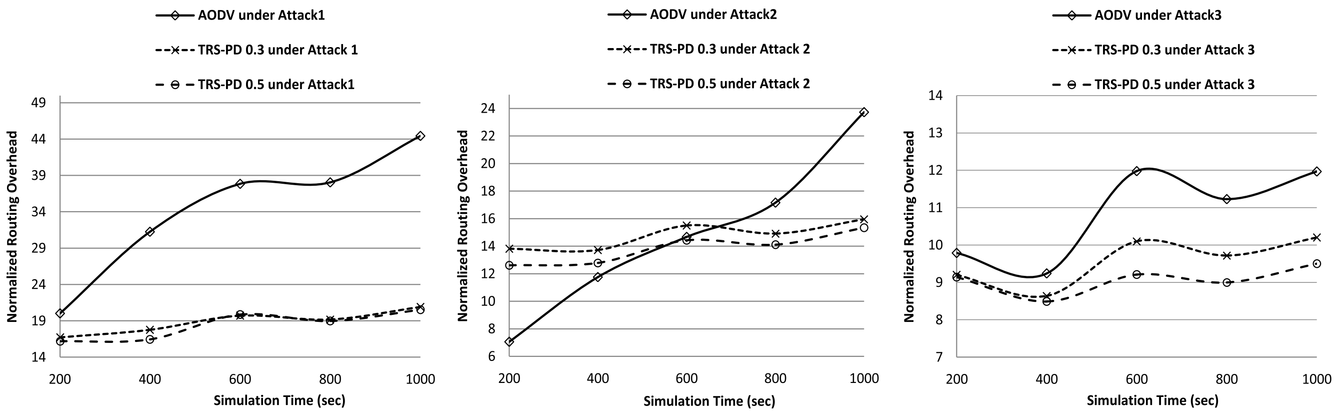


FIGURE 27. PDR vs simulation time with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

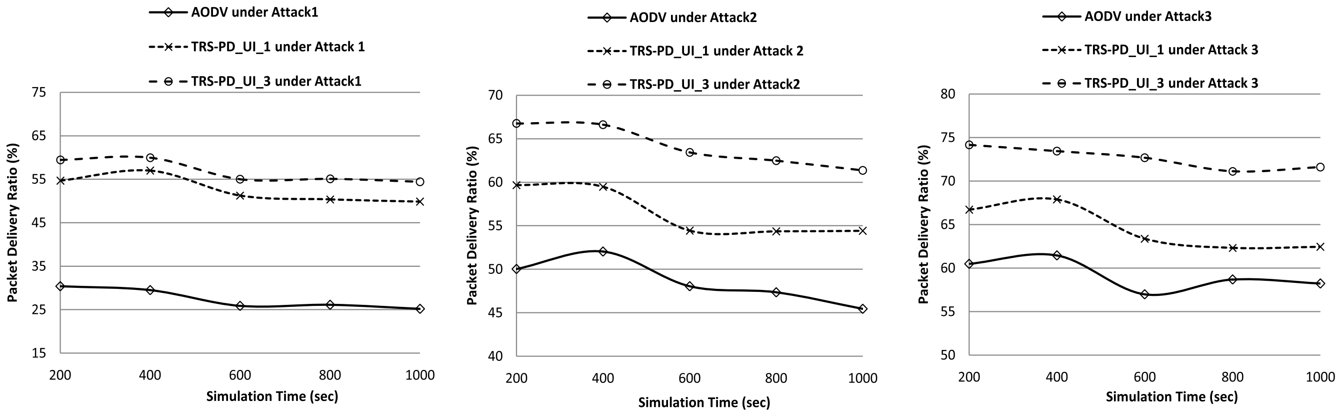
adversaries, the average NRO provided by TRS-PD with $\eta = 0.4$ is significantly better than TRS-PD with $\eta = 0.5$.

Fig. 27 depicts the PDR of TRS-PD under the three adversary models with increasing simulation time by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 27 (a) shows that, under *Attack1*, PDR of TRS-PD with $w1 = 0.3$ decreases from 57.10 to 52.85 and PDR of TRS-PD with $w1 = 0.5$ decreases from 59.96 to 54.40.



(a) (b) (c)

FIGURE 28. NRO vs simulation time with distinct weights. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.



(a) (b) (c)

FIGURE 29. PDR vs simulation time with distinct trust update intervals. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

Fig. 27 (b) shows that, under *Attack2*, PDR of TRS-PD with $w1 = 0.3$ decreases from 63.68 to 59.63 and PDR of TRS-PD with $w1 = 0.5$ decreases from 66.76 to 61.36. Fig. 27 (c) shows that, under *Attack3*, PDR of TRS-PD with $w1 = 0.3$ decreases from 72.55 to 69.12 and PDR of TRS-PD with $w1 = 0.5$ decreases from 74.16 to 71.13. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $w1 = 0.5$ is considerably better than TRS-PD with $w1 = 0.3$.

Fig. 28 depicts the NRO of TRS-PD under the three adversary models with increasing simulation time by taking $w1 = 0.3$ (and $w2 = 0.7$) and $w1 = 0.5$ (and $w2 = 0.5$). Fig. 28 (a) shows that, under *Attack1*, NRO of TRS-PD with $w1 = 0.3$ increases from 16.71 to 20.91, and NRO of TRS-PD with $w1 = 0.5$ increases from 16.18 to 20.53. Fig. 28 (b) shows that, under *Attack2*, NRO of TRS-PD with $w1 = 0.3$ increases from 13.72 to 15.96, and NRO of TRS-PD with $w1 = 0.5$ increases from 12.61 to 15.34. Fig. 28 (c) shows that, under *Attack3*, NRO of TRS-PD with $w1 = 0.3$ increases from 8.64 to 10.20, and NRO of TRS-PD with $w1 = 0.5$ increases from 8.49 to 9.50. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $w1 = 0.5$ is notably better than TRS-PD with $w1 = 0.3$.

Fig. 29 depicts the PDR of TRS-PD under the three adversary models with increasing simulation time by taking $UI = 1$ and $UI = 3$. Fig. 29 (a) shows that, under *Attack1*, PDR of TRS-PD with $UI = 1$ decreases from 56.99 to 49.86 and PDR of TRS-PD with $UI = 3$ decreases from 59.96 to 54.40. Fig. 29 (b) shows that, under *Attack2*, PDR of TRS-PD with $UI = 1$ decreases from 59.68 to 54.35 and PDR of TRS-PD with $UI = 3$ decreases from 66.76 to 61.36. Fig. 29 (c) shows that, under *Attack3*, PDR of TRS-PD with $UI = 1$ decreases from 67.89 to 62.33 and PDR of TRS-PD with $UI = 3$ decreases from 74.16 to 71.13. We can analyze that, under all three adversaries, the average PDR provided by TRS-PD with $UI = 3$ is significantly higher than TRS-PD with $UI = 1$.

Fig. 30 depicts the NRO of TRS-PD under the three adversary models with increasing simulation time by taking $UI = 1$ and $UI = 3$. Fig. 30 (a) shows that, under *Attack1*, NRO of TRS-PD with $UI = 1$ increases from 18.57 to 24.48

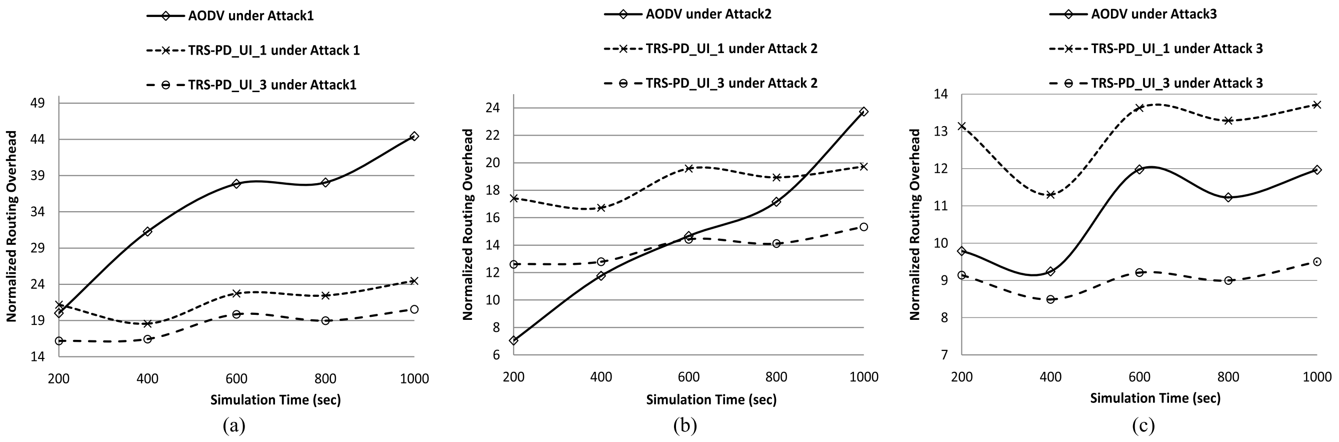


FIGURE 30. NRO vs simulation time with distinct trust update interval. (a) Performance under Attack1. (b) Performance under Attack2. (c) Performance under Attack3.

and NRO of TRS-PD with $UI = 3$ increases from 16.18 to 20.53. Fig. 30 (b) shows that, under *Attack2*, NRO of TRS-PD with $UI = 1$ increases from 16.73 to 19.73, and NRO of TRS-PD with $UI = 3$ increases from 12.61 to 15.34. Fig. 30 (c) shows that, under *Attack3*, NRO of TRS-PD with $UI = 1$ increases from 11.30 to 13.72, and NRO of TRS-PD with $UI = 3$ increases from 8.49 to 9.50. We can analyze that, under all three adversaries, the average NRO provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

E. RESULT ANALYSIS

The tests conducted under different network scenarios prove that: (i) The average PDR and the average NRO provided by TRS-PD with $\eta = 0.4$ is significantly better than TRS-PD with $\eta = 0.5$. (ii) The average PDR and the average NRO provided by TRS-PD with $w1 = 0.5$ is considerably better than TRS-PD with $w1 = 0.3$. (iii) The average PDR and the average NRO provided by TRS-PD with $UI = 3$ is significantly better than TRS-PD with $UI = 1$.

V. CONCLUSION

In our previous work, we devised a novel trust based scheme (viz. TRS-PD) for MANETs in IIoT in order to identify adversaries following different kinds of attack-patterns before they actually launch packet dropping attacks. The scheme intended to isolate the adversaries at an early stage in order to improve the quality-of-services. In this work, we attempt to identify the best choices of values of distinct parameters by carrying out sensitivity analysis of the scheme. The sensitivity analysis is carried out in different network conditions by taking packet delivery ratio and normalized routing overhead as the performance metrics, and varying the values of distrust threshold, trust component's weight and trust update interval.

The results depict that distrust threshold is a critical component of any trust-based scheme which should be set conservatively in order to optimize the detection rate of the scheme. At the same time, both the distrust components (data packet drop ratio and control packet drop ratio) should be given equal importance as the adversaries attempt to trap benign nodes during route discovery process followed by packet dropping misbehaviors during data transmission process. Moreover, trust update interval is a key component which significantly affects the performance of a trust-based scheme as frequent updates in the distrust values may lead to increased false positives/false negatives and unnecessary route alterations. Hence, we can conclude that it is imperative to carry out sensitivity analysis of a security scheme in order to tune the parameter values for different network conditions.

ACKNOWLEDGMENT

The authors would like to thank Prof. (Dr.) Jin Li of Guangzhou University (China), Prof. (Dr.) D.C. Jinwala of Sardar Vallabhbhai National Institute of Technology (India), Charles Perkins of Nokia Research (USA) and Nuno Preguiça, Universidade Nova de Lisboa (Portugal) for their able guidance and motivation. They would also like to thank the anonymous reviewers and the editors.

REFERENCES

[1] R. H. Jhaveri and N. M. Patel, "Attack-pattern discovery based enhanced trust model for secure routing in mobile ad-hoc networks," *Int. J. Commun. Syst.*, vol. 30, no. 7, p. e3148, 2017.

[2] G. Arulkumaran and R. K. Gnanamurthy, "Fuzzy trust approach for detecting black hole attack in mobile adhoc network," in *Mobile Networks and Applications*. Amsterdam, The Netherlands: Springer, 2017, pp. 1–8.

[3] M. Z. A. Bhuiyan, G. Wang, J. Wu, J. Cao, X. Liu, and T. Wang, "Dependable structural health monitoring using wireless sensor networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 14, no. 4, pp. 363–376, Jul./Aug. 2017.

[4] R. H. Jhaveri and N. M. Patel, "Mobile ad-hoc networking with AODV: A review," *Int. J. Next-Generat. Comput.*, vol. 6, no. 3, pp. 165–191, 2015.

[5] S. Peng, A. Yang, L. Cao, S. Yu, and D. Xie, "Social influence modeling using information theory in mobile social networks," *Inf. Sci.*, vol. 379, pp. 146–159, Feb. 2017.

[6] R. H. Jhaveri, S. J. Patel, and D. C. Jinwala, "DoS attacks in mobile ad hoc networks: A survey," in *Proc. 2nd Int. Conf. Adv. Comput. Commun. Technol.*, Jan. 2012, pp. 535–541.

[7] D. M. Shila, W. Shen, Y. Cheng, X. Tian, and X. S. Shen, "AMCloud: Toward a secure autonomic mobile ad hoc cloud computing system," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 74–81, Apr. 2017.

[8] Q. Liu, Y. Guo, J. Wu, and G. Wang, "Effective query grouping strategy in clouds," *J. Comput. Sci. Technol.*, vol. 32, no. 6, pp. 1231–1249, Nov. 2017.

[9] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," in *Cluster Computing*. New York, NY, USA: Springer, 2017, doi: 10.1007/s10586-017-0849-9.

[10] J. Li, L. Sun, Q. Yan, Z. Li, W. Srisa-An, and H. Ye, "Significant permission identification for machine learning based Android malware detection," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2017.2789219.

[11] Q. Liu, G. Wang, F. Li, S. Yang, and J. Wu, "Preserving privacy with probabilistic indistinguishability in weighted social networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 5, pp. 1417–1429, May 2017, doi: 10.1109/TPDS.2016.2615020.

[12] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 105, pp. 117–123, Mar. 2018.

[13] T. A. Ramrekha, O. Adigun, A. Ladas, N. Weerasinghe, and C. Politis, "Towards a scalable routing approach for mobile ad-hoc networks," in *Proc. IEEE 20th Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Sep. 2015, pp. 261–266.

[14] W. Yang, G. Wang, M. Z. A. Bhuiyand, and K. R. Choo, "Hypergraph partitioning for social networks based on information entropy modularity," *J. Netw. Comput. Appl.*, vol. 85, pp. 59–71, May 2017. [Online]. Available: http://dx.doi.org/10.1016/j.jnca.2016.10.002

[15] R. H. Jhaveri, N. M. Patel, and D. C. Jinwala, "A composite trust model for secure routing in mobile ad-hoc networks," in *Ad Hoc Networks*, J. H. Ortiz and A. P. de la Cruz, Eds. Rijeka, Croatia: InTech, 2017, pp. 19–45.

[16] R. H. Jhaveri and N. M. Patel, "Evaluating energy efficiency of secure routing schemes for mobile ad-hoc networks," *Int. J. Next-Generat. Comput.*, vol. 7, no. 2, pp. 130–143, 2016.

[17] R. H. Jhaveri and N. M. Patel, "A sequence number based bait detection scheme to thwart grayhole attack in mobile ad hoc networks," *Wireless Netw.*, vol. 21, no. 8, pp. 2781–2798, 2015.

[18] Z. Chen, L. Peng, C. Gao, B. Yang, Y. Chen, and J. Li, "Flexible neural trees based early stage identification for IP traffic," *Soft Comput.*, vol. 21, no. 8, pp. 2035–2046, 2017.

[19] W. Jiang, G. Wang, M. Z. A. Bhuiyan, and J. Wu, "Understanding graph-based trust evaluation in online social networks: Methodologies and challenges," *ACM Comput. Surv.*, vol. 49, no. 1, p. 10, 2016.

[20] E. Luo, Q. Liu, and G. Wang, "Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks," *IEEE Commun. Lett.*, vol. 20, no. 9, pp. 1772–1775, Sep. 2016.

[21] M. S. Khan, M. I. Khan, S.-U.-R. Malik, O. Khalid, M. Azim, and N. Javaid, "MATF: A multi-attribute trust framework for MANETs," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, p. 197, 2016, doi: 10.1186/s13638-016-0691-4.

[22] T. Gazdar, A. Belghith, and A. Almogren, "DTCF: A distributed trust computing framework for vehicular ad hoc networks," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 3, pp. 1533–1556, 2017.

[23] S. A. Thorat and P. J. Kulkarni, "Uncertainty analysis framework for trust based routing in MANET," *Peer-to-Peer Netw. Appl.*, vol. 10, no. 4, pp. 1101–1111, 2017.

[24] S. Sargunavathi and J. M. L. Manickam, "Design and development of CTSR with direct & indirect observations of MANET applications," *Mobile Netw. Appl.*, vol. 22, no. 4, pp. 712–718, 2017.

[25] X. Li, Z. Jia, P. Zhang, R. Zhang, and H. Wang, "Trust-based on-demand multipath routing in mobile ad hoc networks," *IET Inf. Secur.*, vol. 4, no. 4, pp. 212–232, 2010.

[26] I. Jawhar, F. Mohammed, J. Al Jaroodi, and N. Mohamed, "TRAS: A trust-based routing protocol for ad hoc and sensor networks," in *Proc. IEEE 2nd Int. Conf. Big Data Secur. Cloud (BigDataSecurity), IEEE Int. Conf. High Perform. Smart Comput. (HPSC), IEEE Int. Conf. Intell. Data Secur.*, Apr. 2016, pp. 382–387.

[27] C. A. Kerrache, N. Lagraa, C. T. Calafate, and A. Lakas, "TFDD: A trust-based framework for reliable data delivery and DoS defense in VANETs," *Veh. Commun.*, vol. 9, pp. 254–267, Jul. 2017.

[28] B. Wang, X. Chen, and W. Chang, "A light-weight trust-based QoS routing algorithm for ad hoc networks," *Pervasive Mobile Comput.*, vol. 13, pp. 164–180, Aug. 2014.

[29] W. Li and H. Song, "ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 960–969, Apr. 2016.

[30] P. Sethuraman and N. Kannan, "Refined trust energy-ad hoc on demand distance vector (ReTE-AODV) routing algorithm for secured routing in MANET," *Wireless Netw.*, vol. 23, no. 7, pp. 2227–2237, 2017.

[31] Z. Wu, W. Lin, Z. Zhang, A. Wen, and L. Lin, "An ensemble random forest algorithm for insurance big data analysis," in *Proc. IEEE Int. Conf. Comput. Sci. Eng. (CSE) IEEE Int. Conf. Embedded Ubiquitous Comput. (EUC)*, Jul. 2017, pp. 531–536.

[32] Y.-G. Wang, G. Zhu, and Y.-Q. Shi, "Transportation spherical watermarking," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2063–2077, Apr. 2018.

[33] A. Castiglione *et al.*, "Hierarchical and shared access control," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 850–865, Apr. 2016.

[34] C. Jun, W. Yu, L. Yan, and L. J. Zhen, "Enhancing network capacity by weakening community structure in scale-free network," *Future Generat. Comput. Syst.*, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.future.2017.08.014

RUTVIJ H. JHAVERI received the Ph.D. degree in computer engineering from CHARUSAT University, Changa, India, in 2016. He has been an Assistant Professor with the Department of Computer Engineering, SVM Institute of Technology, Bharuch, India, affiliated to Gujarat Technological University, since 2002. He authored over 65 papers/book-chapters published by prominent publishing houses, such as Wiley, Springer, Elsevier, ACM, IEEE, IET, INTECH, and others. His research interests include resiliency in software defined networking, security in wireless networks, and cyber security. He was a recipient of the Pedagogical Innovation Award from Gujarat Technological University in 2017 for his innovative teaching methodology and research contribution. He serves as an editorial board member/reviewer for various reputed international journals and also as a program committee member/reviewer for renowned international conferences. He possesses the memberships of various technical organizations, such as ISTE, IAENG, IDES, IRED, IACSIT, ICST and others. Apart from this, he also serves as a Committee Member for Bharuch district in SMART VILLAGE PROJECT of Government of Gujarat.

NARENDRA M. PATEL received the B.E. and M.E. degrees from M. S. University, Vadodara, in 1993 and 1997, respectively, and the Ph.D. degree from SVNIT, Surat, in 2012. He is currently an Associate Professor in computer engineering with Birla Vishvakarma Mahavidyalaya, V. V. Nagar, India. He has more than 22 years of academic experience. He authored over 60 papers/book-chapters, which are published in prominent international journals and conference proceedings. He has supervised more than 55 PG dissertations in computer engineering. He has rendered his service as an expert in several workshops, training programs, conferences, and seminars. His research interests include digital image processing, real time operating systems, and distributed systems.

YUBIN ZHONG serves as a Professor in information systems and operation research control with Guangzhou University, China. Besides, he is currently the Dean of the Department of Operation Research and Control, College of Mathematics and Information Science, Guangzhou University. He also shoulders the responsibilities as the Vice President of the China Fuzzy Information and Engineering Association. He has authored or co-authored over 60 papers in renowned international journals. His main research directions are information systems and operational optimization, mathematical theory of super-algebraic structure, and knowledge representation. He serves in the Editorial Board of Management and Fuzzy Mathematics.

ARUN KUMAR SANGAIAH received the master's degree in engineering from Anna University in 2007 and the Ph.D. degree from VIT University, Vellore, India, in 2014. He was a Visiting Professor with the School of Computer Engineering, Nanhai Dongruan Information Technology Institute, China, from 2016 to 2017. He is currently an Associate Professor with the School of Computing Science and Engineering, VIT University. He has authored or co-authored over 130 scientific papers in high standard SCI journals like IEEE-TII, *IEEE Communications Magazine*, IEEE Systems, IEEE-IoT, IEEE TSC, and IEEE ETC. In addition, he has authored/edited over eight books (Elsevier, Springer, Wiley, and Taylor and Francis) and 50 journal special issues, such as *IEEE Communications Magazine*, IEEE-IoT, and *IEEE Consumer Electronics Magazine*. Also, he has registered an Indian patent in the area of computational intelligence. His area of interest includes software engineering, computational intelligence, wireless networks, bio-informatics, and embedded systems. Besides, He is responsible for Editorial Board Member/Associate Editor of various international SCI journals.