

RESEARCH

Open Access



MATF: a multi-attribute trust framework for MANETs

Muhammad Saleem Khan¹, Majid Iqbal Khan¹, Saif-Ur-Rehman Malik¹, Osman Khalid², Mukhtar Azim¹ and Nadeem Javaid^{1*}

Abstract

To enhance the mobile ad hoc networks (MANETs) security, various trust-based security schemes have been proposed. However, in most of the trust-based security schemes, a node's trust is computed based on a single trust attribute criteria, such as data forwarding. Using single trust attribute criteria may cause the bootstrapping problem, which refers to the time required by the trust-based scheme to build trust and reputation among nodes in the network. The bootstrapping problem in these schemes may provide more opportunities to misbehaving nodes to drop packets and remain undetected for longer time in the network. Moreover, using single trust attribute criteria does not effectively deal with the selective misbehavior by a smart malicious node.

In this work, we propose a scheme that is based on the multi-attribute trust criteria to minimize the bootstrapping time, which ultimately improves the performance of the scheme in terms of high malicious node detection rate, low false-positive rate, and packet loss rate. The contributions of this paper are (a) identification of trust attributes along with the development of a comprehensive multi-attribute trust framework (MATF) using multiple watchdogs for malicious node identification and isolation, (b) formal modeling and verification of our proposed MATF using HLPN, SMT-Lib, and Z3 Solver, and (c) simulation-based validation and evaluation of the proposed trust framework in the context of optimized link state routing (OLSR) protocol against various security threats, such as message dropping, message modification, and link withholding attacks. The simulation results revealed that the proposed trust framework achieves about 98 % detection rate of malicious nodes with only 1–2 % false positives. Moreover, the proposed MATF has an improved packet delivery ratio as compared to the single attribute-based scheme.

Keywords: Trust, Packet loss, Bootstrapping time, Security attacks, Formal verification, High-level Petri nets

1 Introduction

Due to the non-availability of central authority and the unreliability of wireless links, the routing protocols in mobile ad hoc networks (MANETs) are vulnerable to various types of security threats [1]. The resource-constrained nature of MANETs with continuously evolving topology and frequent network partitioning complicates the security challenges in MANETs' routing. Most of the secure routing protocols for MANETs utilize some form of cryptography to ensure the network security [2–4]. However, there are scenarios, where cryptography techniques fail to capture malicious behavior of a node. For example, (a) to

disrupt the network topology, a node may provide falsified routing information to other nodes, (b) to preserve the battery, a node may not participate in the routing functions, and (c) a node may drop data packets instead of forwarding because of the malicious intention. To address these issues, trust-based security schemes [5–10] have been proposed to augment the security of traditional cryptography-based approaches.

In MANETs, trust can be defined as to what extent a node can fulfill the expectations of other node(s) as per the specification of an underlying communication protocol [11]. In trust-based security schemes, each node within the network manages an independent trust table to compute and store the trust values of other nodes. The routing decisions are based on the computed trust values of the nodes. Although a lot of research work has been carried in the field of trust and reputation based systems in

*Correspondence: nadeemjavaid@comsats.edu.pk

¹Department of Computer Sciences, COMSATS Institute of Information Technology, Islamabad, Pakistan

Full list of author information is available at the end of the article

MANETs, however, almost all the proposed schemes suffer from one basic problem known as bootstrapping problem [12]. It refers to the time required by the trust-based scheme to build trust and reputation among nodes in the network. Such delay in accumulation of trust and reputation is often not acceptable in time-critical applications. Due to the slow trust building process, a misbehaving node may have more opportunities to drop packets before being detected as malicious. One of the basic reasons for the aforementioned bootstrapping problem is that in most of the trust-based security schemes, an evaluated node's trust is computed based on a single trust attribute, such as data forwarding [13–17]. Moreover, using single trust attribute may not effectively deal with the problem of selective misbehavior [12]. A smart malicious node may misbehave in the context of one network function and behave properly for other network functions. For example, a node may misbehave in the context of data forwarding while demonstrating good behavior when dealing with the control packet forwarding. As the existing schemes [7–10, 13–17] use single trust attribute, the aforementioned selective misbehaving node is declared as malicious node and isolated from the routing path, hence no longer will be available to be used for other network functions.

In trust-based security schemes, each node collects two major types of information about other nodes: first-hand information (based on self-observations) and second-hand information (based on the other node observations). In literature, efforts have been made to minimize the bootstrapping time and to increase the detection rate by using second-hand information to evaluate the trustworthiness of the nodes [14, 17]. However, the aforementioned schemes still suffer from data sparsity problem [14]. In trust-based security schemes, data sparsity is a situation where lack of information or insufficient interaction experience makes it difficult to evaluate the node's trust, especially in the early time of network establishment. Moreover, using second-hand information without any filtration may cause bad-mouthing and false praise attack [11], which ultimately cause high false positive and false negative rate. In bad-mouthing attack, a misbehaving node propagates dishonest and unfair recommendations against an innocent node with a negative intention to confuse the trust model. Similarly, in false praise attack, a misbehaving node propagates unfairly positive recommendations against the malicious node to mislead the trust model.

It is also of critical importance to prove the correctness of the trust-based security schemes in dynamic and unpredictable environments, such as of MANETs. A well-established approach to prove the correctness of a system's model is by employing a formal verification process [18].

To minimize the bootstrapping time and expedite the trust building process, and to effectively deal with the

selective misbehavior, there is a strong need for a mechanism that works on multi-attribute-based trust strategy. Each node should be observed in the context of all the possible network functions, such as control message generation, control message forwarding, and data packet forwarding. Moreover, an efficient recommendation filtration technique is required to filter the source of information and information itself. To avoid bad-mouthing and false praise attacks, second-hand information from only designated and trustworthy nodes must be considered in a trust computation process.

Our contributions: In this work, we address the bootstrapping and delusive trust dissemination problem when using second-hand information. We propose a trust-based security scheme which uses multi-attribute-based trust criteria, such as control packet generation, control packet forwarding, and data packet forwarding. Using multi-attribute trust criteria which minimizes the bootstrapping time and expedite the trust building process, as nodes are assessed in the context of different aforementioned network functions. Moreover, to avoid bad-mouthing and false praise attacks, second-hand information is considered from only those nodes called *watchdog nodes*, whose trust values are above some threshold. Furthermore, second-hand information from recommender nodes with trust deviation (τ_{dev}) value¹ less than the deviation threshold (τ_{dev-th}) will be considered in the trust computation process. This paper has the following major contributions.

- Identification of the trust attributes for a node's trust building process.
- Development of a comprehensive multi-attribute and multiple watchdog nodes trust framework (MATF) for malicious node detection and isolation.
- Formal verification of our proposed MATF using high-level Petri nets (HLPNs), satisfiability modulo theories-library (SMT-Lib), and Z3 Solver.
- Implementation of the proposed trust framework in the context of optimized link state routing (OLSR) protocol in NS-2 [19].
- Simulation-based validation and evaluation of the proposed MATF in comparison with the recently proposed trust scheme by Shabut et al. [14] (single trust-attribute-based scheme), against various security threats, such as message dropping, message modification, and link withholding attacks.
- Security analysis of the proposed MATF.

The rest of the paper is organized as follows. In Section 2, we present the related work. Section 3 presents the discussion on trust and its formulation in MANETs along with a multi-attribute trust framework. Section 4 presents the formal modeling and verification of the proposed framework. Section 5 presents the simulation

results and summarizes the performance evaluation of the proposed model. Security analysis of the proposed scheme is presented in Section 6, and the paper is concluded in Section 7.

2 Related work

Trust-based security schemes are one of the active research areas for ensuring the security in MANETs [20]. In recent years, different trust-based security schemes have been proposed to enhance the security in MANETs. In these schemes, nodes evaluate their neighbor nodes based on the first-hand information or using recommendations from other nodes [12, 20]. Though, these schemes paid some attention to the problem of bootstrapping and delusive trust dissemination problem, however, an efficient mechanism to mitigate the aforementioned problem is still a challenging issue in MANETs. We categorize the state-of-the-art schemes in the following categories.

2.1 Watchdog and path-rater schemes

One of the key works in trust-based schemes was presented by Marti et al. [13]. They proposed *watchdog and path-rater* mechanisms implemented on the dynamic source routing (DSR) protocol to minimize the impact of malicious nodes on the throughput of the network. The aforementioned approach detects the misbehaving nodes by using only source node as a monitoring node. However, the proposed scheme has some major shortcomings, such as it cannot detect the misbehaving nodes in the case of ambiguous collision, receiver collision, limited transmission power, partial dropping, and collaborative attacks [21]. Moreover, watchdog and path-rater mechanisms utilize only first-hand information for node misbehavior detection that causes the aforementioned issues.

2.2 Feedback-based schemes

To solve the issues in the watchdog and path-rater schemes, various approaches were proposed, such as acknowledgment-based detection systems including two network-layer acknowledgment-based schemes, termed as TWOACK [22], adaptive acknowledgment (AACK) [23], and enhanced adaptive acknowledgment (EEACK) [21]. The TWOACK scheme has focused to solve the receiver collision and limited transmission power problems of the watchdog and path-rater approach. Every data packet transmitted is acknowledged by every three consecutive nodes along the path from the source to the destination. Sheltami et al. [23] proposed an improved version of the acknowledgment-based scheme, AACK. The AACK is the intrusion detection system which is a combination of TWOACK and end-to-end acknowledgment scheme. Although, AACK has significantly reduced the overhead as compared to TWOACK scheme, it still suffers from the problem of detecting malicious nodes

generating false misbehavior report and forged acknowledgment packets. To remove the shortcomings of the acknowledgement-based schemes, Shakshuki et al. [21] proposed EAACK protocol to detect misbehavior nodes in MANETs' environment using digital signature algorithm (DSA) [24] and Rivest-Shamir-Adleman (RSA) algorithm [25] digital signatures. Although, their technique can validate and authenticate the acknowledgement packets, yet at the expense of extra resources, and it also requires pre-distributed keys for digital signatures.

2.3 Network monitoring-based schemes

Buchegger et al. presented a cooperation of nodes-fairness in distributed ad-hoc networks (CONFIDANT) protocol [7] to detect misbehaving nodes in the network. In addition to first-hand information, second-hand information is also used while computing a node's trustworthiness. In CONFIDANT protocol, first-hand information is propagated after every 3 s, while weight given to the second-hand information is 20 %. To avoid false praise attack [11], only negative experiences as second-hand information are shared among nodes. One of the shortcomings in CONFIDANT protocol is that ALARM messages used in the protocol can be exploited by the bad-mouthing nodes. Bad-mouthing nodes may generate ALARM messages against the legitimate nodes to induce biasness in the protocol's results [22]. Similarly, a collaborative reputation mechanism to enforce node cooperation in MANETs called CORE [9] also uses the second-hand information to compute the reputation of a node. Only positive experiences are shared by the node with other nodes in the network to avoid bad-mouthing attack.

In contrast to CONFIDANT and CORE [9], observation-based cooperation enforcement in ad hoc networks (OCEAN) protocol [26] uses only first-hand observation to avoid false praising and bad-mouthing type of attack. In OCEAN, avoid-list strategy is implemented to not forward the traffic from misbehaving nodes. However, if a node identifies that its ID is inserted to the avoid-list, it may change its strategy. A tamper-proof hardware is required to secure the avoid-list to avoid the aforementioned incident.

To filter the second-hand information, [14] proposed a defence trust scheme based on three parameters: (a) confidence value, indicating how many interactions took place between a recommender node and an evaluated node, (b) deviations in the opinions of recommender node and evaluating node, and (c) closeness value, indicating the distance-wise close of recommender node and the evaluating node. On the basis of the aforementioned values, an evaluating node filters the second-hand information in the proposed trust scheme. However, the second-hand information filtration mechanism in the proposed scheme may not work well in some scenario. For example,

recommender nodes $R_1, R_2 \dots R_N$ send the bad reputation value of misbehaving node M to evaluating node E , while node E has a good reputation value about node M based on its own first-hand information. In the aforementioned proposed scheme [14], such recommendations are filtered out because of more deviation in the trust values. In contrast, our proposed MATF scheme filters the recommendation by using the following methodology. When recommendations received at the evaluating node from the recommender node about some particular evaluated node, the evaluating node averages the recommendations already received from all the watchdog nodes (recommender nodes) then, finds the trust deviation of the recommender node's trust value from the average trust value. If the deviation in trust values is less than certain deviation threshold, weight is given to the recommendations in the trust computation; otherwise, no weight is given to these recommendations.

Li et al. [27] proposed a simple trust model which takes into account the packet forwarding ratio as metric to evaluate the trustworthiness of neighbor nodes. A node's trust is computed by the weighted sum of packet forwarding ratio. To find a path trust, continued product of node's trust values in a routing path is computed. The aforementioned approach only considers packet forwarding behavior as a trust metric. A trust prediction model based on the node's historical behavior called trust-based source routing (TSR) protocol was presented in [28]. On the basis of assessment and prediction results, the nodes can select the shortest trusted route to transmit the required packets. One of the weaknesses of this work is that no second-hand information is considered for trust computation that may result in bootstrapping and data sparsity problem [14]. Trust-based security schemes like [16] only consider the security of data traffic, while schemes like [29, 30] only consider the security of control traffic. Moreover, the aforementioned solutions result in more energy consumption due to excessive information propagation and detection messages. In [31], energy efficiency is considered as one of the parameters and have improved previously existing trust-based algorithms.

To summarize, the trust-based security schemes discussed in this section have some open problems that need to be solved. Most of the existing schemes use single trust criteria for the trust building process that causes the bootstrapping and data sparsity problem. Minimizing the bootstrapping time and the data sparsity problem is still an open issue [12, 14]. Moreover, using all the available information from each and every node in the network does help in building reputation and trust among nodes quickly, but as discussed earlier, it makes the system vulnerable to false report attacks. To solve the aforementioned false praise and bad-mouthing attacks, there should be a mechanism which filters the spurious

second-hand information. Although, the aforementioned approaches suggest the misbehavior detection schemes, these schemes use single trust attributes like data forwarding. Moreover, second-hand information are considered from recommender nodes without any filtration that can result in erroneous trust estimation, especially under high nodal mobility. In contrast, our proposed MATF is based on multiple trust attributes with multiple observer nodes that results in better trust estimation. Second-hand information are considered from recommender nodes with deviation values less than the deviation threshold, which results in better trust estimation, especially under high nodal mobility.

3 MATF: the proposed scheme

In this section, we present the trust attributes, trust formulation in the proposed MATF, a mechanism for trust deviation test, and watchdog node selection process. In the proposed MATF, the watchdog node is the designated neighbor node of the evaluating node to monitor the activities of the evaluated node B on the basis of defined trust attributes and is represented by W . It can be the evaluating node itself or any other node that has been assigned the monitoring task by the evaluating node. The evaluating node computes the final trust of the evaluated node based on its own observations and those reported by the watchdog nodes. Our proposed trust model consists of three steps. The first step is the monitoring step, in which an evaluating node S and watchdog nodes W_n observe the behavior of an evaluated node B in the context of trust attributes ρ . For clarity, in the following equations, we treat an evaluating node as one of the watchdog nodes. In the second step, an evaluating node aggregates its own observations and the watchdog nodes' observations in the context of each trust attribute. Finally, an evaluating node computes the final trust of an evaluated node in the context of all the trust attributes using the weighted sum. Also, the value range of ρ is $[0, 1]$, 0 being the minimum and 1 the maximum.

3.1 Trust attributes

Trust attributes are the factors responsible for shaping the trust levels and denoted by ρ . Each trust attribute value ranges between 0 and 1. Before going into the details as how we applied trust in MANETs, first, we discuss the basic trust attributes and then, define our trust model.

We have identified the following trust attributes in the context of control and data traffic for the proposed trust model.

3.1.1 Control packet generation (ρ_{cpg})

Control packet is the protocol-specific information that nodes exchange to build routes and maintain topology. By using this trust attribute, an evaluating node assesses

the trustworthiness of the evaluated node in the context of control packet generation behavior as specified in the underlying routing protocol. Observations of a node W about node B in terms of control packet generation is given in the following equation:

$$\rho_{\text{cpg}}^{W,B}(t, t + \Omega) = \frac{p}{p_{\text{exp}}}, \quad (1)$$

where t is the current time, Ω is the trust update period, p is the total actual number of control messages generated in the time interval $(t, t + \Omega)$ by node B as observed by W , and p_{exp} is the expected number of control messages that should have been generated by node B .

An evaluating node then aggregates its observations and the observations reported by the watchdog nodes to build a reputation about node B as shown in the following equation:

$$\rho_{\text{cpg}}(t, t + \Omega) = \alpha \rho_{\text{cpg}}^{S,B} + (1 - \alpha) \left(\frac{1}{n} \sum_{i=1}^n \rho_{\text{cpg}}^{W_i,B} \right), \quad (2)$$

where α is the weight factor given to an evaluating node observation and watchdog node observations.

3.1.2 Control packet forwarding (ρ_{cpf})

Nodes in a MANET depend on mutual cooperation to forward traffic. A non-cooperative forwarding node may drop packet or forward control packet with delay that can result in the inconsistent view of the network topology. Let us denote the packets that are successfully overheard as p_{ack} . The observations of a node W regarding node B in terms of control packet forwarding can be computed using following equation:

$$\rho_{\text{cpf}}^{W,B}(t, t + \Omega) = 1 - \frac{p - p_{\text{ack}}}{p}. \quad (3)$$

According to the above equation, the minimum possible packet loss rate observed at an evaluating/watchdog node W is 0, while the maximum possible packet loss rate is equal to 1, i.e., all the sent packets are dropped by misbehaving nodes. An evaluating node then aggregates its own observations and that of watchdog nodes to obtain an aggregated reputation of node B in terms of control packet forwarding as follows:

$$\rho_{\text{cpf}}(t, t + \Omega) = \alpha \rho_{\text{cpf}}^{S,B} + (1 - \alpha) \frac{1}{n} \sum_{i=1}^n \left(\rho_{\text{cpf}}^{W_i,B} \right), \quad (4)$$

where α is the weight factor given to an evaluating node observations and watchdog node observations in the above equation.

3.1.3 Data packet forwarding (ρ_{dpf})

In addition of control traffic, nodes are also responsible of relaying data packets. A node may drop the data packet

and forward data packets with delay or with maliciously modified contents. The observations of node W regarding node B in terms of data packet forwarding can be computed using the following equation:

$$\rho_{\text{dpf}}^{W,B}(t, t + \Omega) = 1 - \frac{\xi - p_{\text{ack}}}{\xi}, \quad (5)$$

where ξ is the total number of data packet sent and p_{ack} is the data packet successfully overheard at watchdog node W . Aggregating evaluating node's and watchdog node's observations, we get the aggregated reputation of an evaluated node in the context of data packet forwarding as given in the following equation:

$$\rho_{\text{dpf}}(t, t + \Omega) = \alpha \rho_{\text{dpf}}^{S,B} + (1 - \alpha) \frac{1}{n} \sum_{i=1}^n \left(\rho_{\text{dpf}}^{W_i,B} \right). \quad (6)$$

Algorithm 1 The MATF trust computation algorithm

```

1: procedure MONITORING()
2:    $\rho_{\text{cpg}}^{W,B}(t, t + \Omega) = \frac{p}{p_{\text{exp}}}$ 
3:    $\rho_{\text{cpf}}^{W,B}(t, t + \Omega) = 1 - \frac{p - p_{\text{ack}}}{p}$ 
4:    $\rho_{\text{dpf}}^{W,B}(t, t + \Omega) = 1 - \frac{\xi - p_{\text{ack}}}{\xi}$ 
5:    $\tau_{\text{dev}} = \left| \left( \frac{1}{k-1} \sum_{i=1}^{k-1} \tau_{(W_i,j)} \right) - \tau_{(W_k,j)} \right| \leq \tau_{\text{dev-th}}$ 
6:   Trust_Computation( $\rho_{\text{cpg}}, \rho_{\text{cpf}}, \rho_{\text{dpf}}$ )
7: end procedure
8: procedure TRUST_COMPUTATION( $\rho_{\text{cpg}}, \rho_{\text{cpf}}, \rho_{\text{dpf}}$ )
9:    $\tau_S^B(t, t + \Omega) = \frac{\delta \rho_{\text{cpg}} + \beta \rho_{\text{cpf}} + \gamma \rho_{\text{dpf}}}{\delta + \beta + \gamma}$ 
10: end procedure
11: procedure DECIDE( $nb$ )
12:   if ( $\tau_B < \eta$ ) then //  $\eta$  is threshold for malicious node isolation
13:     isolate( $B$ )
14:     relay_computation()
15:   end if
16: end procedure

```

3.2 Trust formulation and algorithm

We are now able to combine the equations introduced so far into our mathematical model for the multi-attribute trust computation. By combining Eqs. 2, 4 and 6, we obtain

$$\tau_S^B(t, t + \Omega) = \frac{\delta \rho_{\text{cpg}} + \beta \rho_{\text{cpf}} + \gamma \rho_{\text{dpf}}}{\delta + \beta + \gamma}, \quad (7)$$

where δ , β , and γ are weight factors assigned to each metric and $\delta + \beta + \gamma = 3$. The weights can be tuned based on the specific security goal to be achieved. For example, if a higher throughput and packet delivery is concerned, we consider the data traffic as vital, so data forwarding parameter carry more weight than other parameters, such

as control packet generation and forwarding. An evaluating node S aggregates the trust computed for evaluated node B during the time interval $(t, t + \Omega)$ in the context of each trust attribute ρ and assigns weights to each aforementioned attributes in the above equation. The trust computed in Eq. (7) is compared with a threshold value to make a decision regarding trustworthiness of a node.

Algorithm 1 presents the pseudo code for the MATE. In the proposed algorithm, an evaluating node and designated watchdog nodes observe the evaluated node in terms of different network functions during the monitoring period (lines 1–4). A filtration criteria is applied on the recommendations received from watchdog nodes (line 5). Based on the filtered recommendations, an evaluating node computes the trust of an evaluated node (lines 8–10). If the trust of an evaluated node is lower than a threshold (lines 12–13), it is isolated from the routing path and a new route selection process is initiated (Line 14).

3.3 Trust deviation

The trust computed by the watchdog nodes will be used as a second-hand information in the proposed scheme. To avoid bad-mouthing and false praise attacks, only those information will be used by the evaluating node which is received from the designated nodes and have a trust deviation value less than the deviation threshold. Trust deviation can be computed as given in the following equation.

$$\tau_{\text{dev}} = \left| \left(\frac{1}{k-1} \sum_{i=1}^{k-1} \tau_{(W_{i,j})} \right) - \tau_{(W_{k,j})} \right| \leq \tau_{\text{dev-th}}, \quad (8)$$

where $\tau_{(W_{i,j})}$ is the average trust already received from the watchdog node W_i about the evaluated node j and $\tau_{(W_{k,j})}$ is the trust recommendation received from watchdog node W_k about the evaluated node j .

3.4 Watchdog selection process

In order to avoid the bad-mouthing and false praise attacks, the second-hand information in the proposed MATE is considered from only designated and trustworthy watchdog nodes, as discussed in the previous subsection. In this section, we discuss the selection process of the watchdog nodes, which will perform the monitoring task. When the network is initialized, each evaluating node selects a set of neighboring nodes called *watchdog set* to monitor the behavior of a particular evaluated node. The proposed security scheme allows flexibility in the watchdog selection. Depending on the available network topology, one or multiple watchdogs may be selected. There is no fixed ratio per node of watchdog nodes to be selected. It will be varying depending on the available network topology. It is worth mentioning that in case of any change in network topology, an evaluating node will recompute the watchdog nodes. The criteria and selection

Algorithm 2 Relay and watchdog node selection

Description: For a node S , the 1-hop neighbor set $N_1(S)$ and the 2-hop neighbor set $N_2(S)$ are known. $N_1^*(S)$ and $N_2^*(S)$ are the unprocessed 1-hop and 2-hop neighbor sets of node S , respectively. Malicious set(S) is the set of malicious nodes maintained at node S .

Input: $N_1(S)$ and $N_2(S)$.

Output: The sets *relay*(S) and *watchdog*(S).

```

1: watchdog( $S$ )  $\leftarrow \emptyset$ 
2: relay( $S$ )  $\leftarrow \emptyset$ 
3: Malicious set( $S$ )  $\leftarrow \emptyset$ 
4: for each node do  $\tau_{\text{initial}} \leftarrow 1$ 
5: end for
6: procedure RELAY_COMPUTATION()
7:   for each isolated node  $y \in N_2^*(S)$  do
8:     if ( $\text{node } x \in N_1^*(S) \wedge \text{node } x \notin \text{Malicious set}(S)$ 
9:        $\wedge \text{node } x$  is the only neighbor of node  $y$ ) then
10:      relay( $S$ )  $\leftarrow \text{node } x$ 
11:      Remove all the nodes in  $N_2^*(S)$  covered by node  $x$ 
12:    end if
13:  end for
14:  while  $N_1^*(S) \neq \emptyset$  do
15:    if ( $\text{node } x \in N_1^*(S) \wedge \text{node } x \notin \text{Malicious set}(S)$ 
16:       $\wedge \text{node } x$  reaches the maximum number of nodes in  $N_2^*(S)$ ) then relay( $S$ )  $\leftarrow \text{node } x$ ;
17:    Remove all nodes from  $N_1^*(S)$  covered by node  $x$ 
18:  end while
19: end procedure
20: procedure WATCHDOG_COMPUTATION()
21:  while relay( $S$ )  $\neq \emptyset$  do
22:    for each node  $x \in \text{relay}(S)$  do
23:      watchdog( $S$ ) =  $N_1^*(S) \cap N_1(x) \wedge (\tau_{nb} > \eta)$  // common neighbors of evaluating node  $S$  and relay node  $x$ , having trust greater than trust threshold
24:    end for
25:  end while
26: return watchdog( $S$ )
27: end procedure

```

process of watchdog nodes are presented in Algorithm 2, which is a modified version of the relay node selection algorithm presented in [32]. An example scenario of the detailed working of the watchdog selection algorithm is presented below.

In the given scenario, node S discovers its neighbors through exchange of control messages and calculates the one-hop neighbor set N_1 and the two-hop neighbor set

N_2 (used as an input in Algorithm 2). From the set N_1 , each evaluating node S computes the relay node set $R(S)$ (lines 6–18) and the watchdog set $W(S)$, having a trust value greater than the trust threshold (lines 20–26). $R(S)$ is the smallest possible subset of $N_1(S)$ required to reach all nodes in $N_2(S)$.

As an example, in Fig. 1, $R(S) = \{B, C, E\}$ contains the minimum number of one-hop neighbors of S required to reach all two-hop neighbors of S . Thereafter, node S selects the watchdog set for each node present in $R(S) = \{B, C, E\}$. To calculate the watchdog set for each node in the $R(S)$, the node S takes the intersection of the one-hop neighbor set $N_1(S)$ and the one-hop neighbor set of each relay node.

Node S broadcasts the $W(S)$ to the neighboring nodes by appending it in the periodic control messages along with $R(S)$. This enables the neighboring nodes of S to check whether or not they have been selected as a watchdog. By utilizing the broadcast information sent by the node S , each node builds the watchdog selector set. The watchdog selector set consists of all those nodes that have selected node W as a watchdog. For example, as reflected in Fig. 1, node S populates the $W(S)\{A, H\}$ for the relay node C by taking the intersection of sets $N_1(S) = \{A, B, C, D, E, H\}$ and $N_1(C) = \{A, S, H, X\}$. Thereafter, node S broadcasts the watchdog set to inform both nodes A and H that from now onward, these nodes have to monitor node C .

4 Formal modeling and verification of the MATF

Formal verification is the process verifying that algorithms work correctly with respect to some formal property [33]. Formally modeling systems helps to analyze the interconnection of components and processes and how the information is processed in the system [34]. Formal modeling provides valuable tools to design, evaluate, and

verify such protocols [35]. To verify the correctness of the MATF, we use HLPNs for the modeling and analysis [18]. HLPNs provide a mathematical representation and help to analyze the behavior and structural properties of the system.

To perform a formal verification of the MATF, the HLPN models are first translated into SMT-Lib [36] using the Z3 Solver [37]. Then, the correctness properties were identified and verified to observe the expected behavior of the models.

In this section, we present a brief overview of HLPNs and a formal verification of the MATF.

4.1 High-level Petri nets

Petri nets are used to model systems which are non-deterministic, distributed, and parallel in nature. HLPNs are a variation of conventional Petri nets. A HLPN is a structure comprised of a seven-tuple, $N = (P, T, F, \varphi, R, L, M_0)$. The meaning of each variable is provided in Table 1.

4.2 SMT-Lib and Z3 Solver

SMT is an area of automated deduction for checking the satisfiability of formulas over some theories of interest and has the roots from Boolean satisfiability solvers (SAT) [34]. The SMT-Lib is an international initiative that provides a standard benchmarking platform that works on common input/output framework. In this work, we used Z3, a high-performance theorem solver and satisfiability checker developed by Microsoft Research [38].

4.3 Modeling and verification of the MATF

To model and verify the design of the MATF, the places P and the associated types need to be specified. The data type refers to a non-empty set of data items associated with a P . The data types used in the HLPN model of the MATF are described in Table 2. Figure 2a present the HLPN model for the relay and watchdog node selection

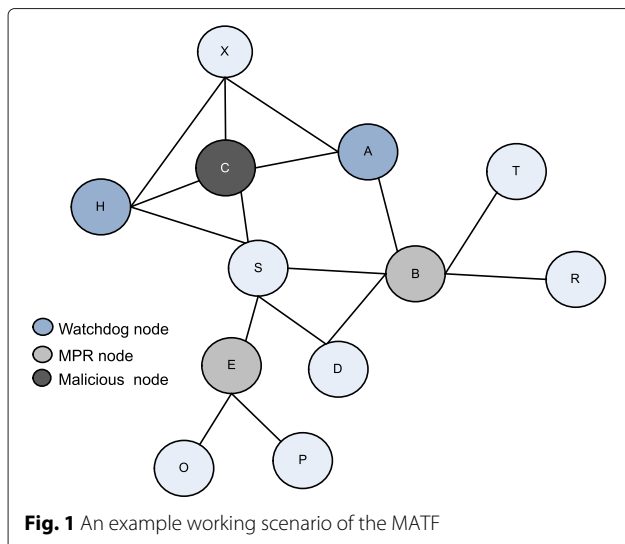


Fig. 1 An example working scenario of the MATF

Table 1 Variables and meaning

Variables	Meanings
P	A finite set of nodes, called <i>Places</i>
T	Set of transitions, that is, $P \cap T = \emptyset$
F	A set of directed edges called arcs, known as the flow relation, that is, $F \subseteq (P \times T) \cup (T \times P)$
φ	A mapping function used to map P to data types
R	A transition rule mapping T to predicate logic formula, that is, $R : T \rightarrow \text{Formula}$
L	A label mapping F to labels, that is, $L : F \rightarrow \text{Label}$
M_0	An initial marking of places with tokens, that is, $M : P \rightarrow \text{Tokens}$

Table 2 Data types and their descriptions

Types	Description	Types	Description
sn_id	Source node ID	Status	Message status
msg	Message	Trust values	Trust observations
wd_id	Watchdog node ID	Trust_threshold	Trust-threshold
relay_id	Relay node ID	Msg-type	Message type
relay-msg	Message sent by relay node	Timer	TC packet timer

in the MATF. Moreover, message forwarding, trust computation, and malicious node isolation are depicted in the HLPN model shown in Fig. 2b. As depicted in Fig. 2a, there are six places in relay and watchdog selection HPLN, whereas seven places in the HLPN model for trust

computation, as shown in Fig. 2b. The names of places and description are given in Table 3. The next step is to define the set of rules, pre-conditions, and post-conditions to map to T . The mapping of transition T to the processes used in the MATF, referred to as rules (R). After defining the notations, we can now define formulas (pre- and post-conditions) to map on transitions in the following. The set of transitions $T = \{ \text{Gen-Nlist}, \text{Gen-WDN}, \text{Gen-relay}, \text{Broadcast}, \text{Forward}, \text{Trust-obs}, \text{Comp-Mali} \}$. The following are the rules used for modeling and verification.

The rule R1 depicts the HELLO message processing. When the network is initialized, nodes exchange the HELLO messages with each other to discover the neighbors in the network. The HELLO message contains the list of one-hop neighbors of a node. On the basis of

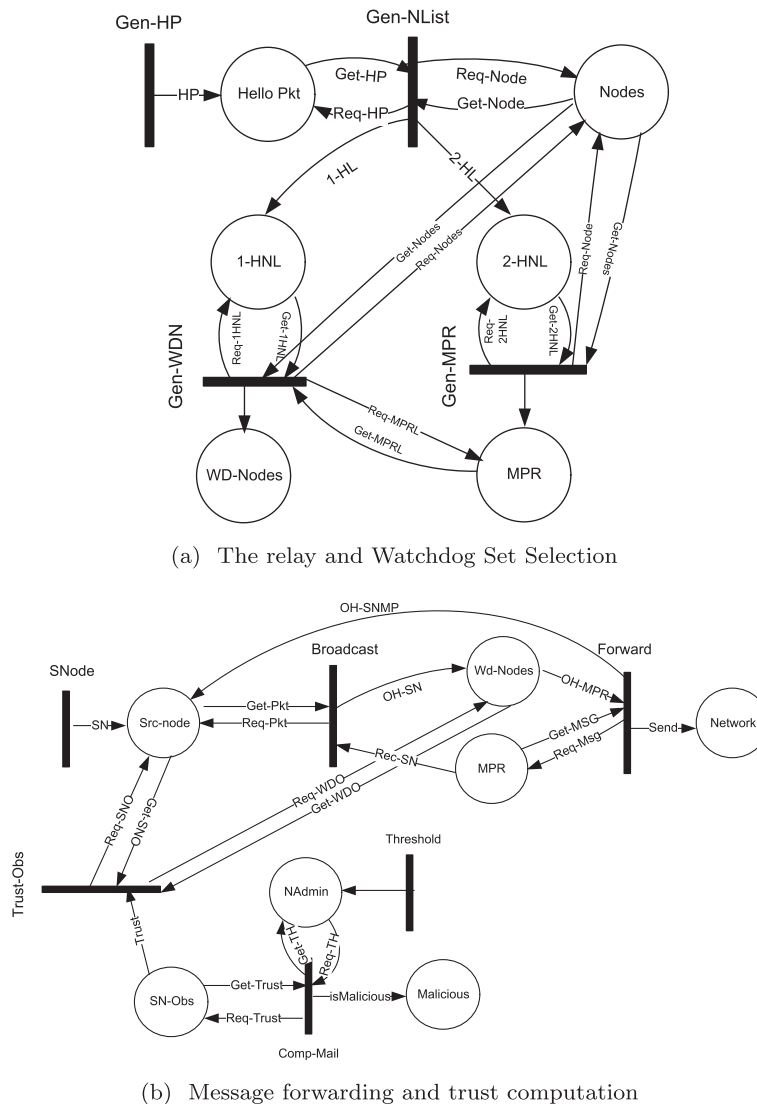
**Fig. 2** HLPN of the MATF (a, b)

Table 3 Places and mappings of data types to the places

Places	Mappings
$\varphi(\text{Hello Pkt})$	$\mathbb{P}(\text{sn_id} \times \text{msg-type} \times 1\text{-HNL} \times \text{timer})$
$\varphi(\text{Nodes})$	$\mathbb{P}(\text{sn_id} \times \text{nb_id} \times \text{timer})$
$\varphi(1\text{-HNL})$	$\mathbb{P}(\text{sn_id} \times \text{nb_id} \times \text{nb_id} \times \text{status} \times \text{timer})$
$\varphi(2\text{-HNL})$	$\mathbb{P}(\text{sn_id} \times \text{msg} \times \text{nb_id} \times \text{nb2hop_id} \times \text{timer})$
$\varphi(\text{Source Node})$	$\mathbb{P}(\text{sn_id} \times \text{msg} \times \text{wd_id} \times \text{relay_id} \times \text{msg-type} \times \text{relay-msg} \times \text{timer})$
$\varphi(\text{WD-Nodes})$	$\mathbb{P}(\text{sn_id} \times \text{msg} \times \text{wd_id} \times \text{relay_id} \times \text{msg-type} \times \text{relay-msg} \times \text{timer})$
$\varphi(\text{relay-nodes})$	$\mathbb{P}(\text{sn_id} \times \text{msg} \times \text{wd_id} \times \text{relay_id} \times \text{msg-type} \times \text{timer} \times \text{status})$
$\varphi(\text{Network})$	$\mathbb{P}(\text{sn_id} \times \text{msg} \times \text{relay_id} \times \text{msg-type})$
$\varphi(\text{SN-Obs})$	$\mathbb{P}(\text{trust-value-fwd} \times \text{trust-value-gen} \times \text{trust-value-mod} \times \text{relay_id})$
$\varphi(\text{NAdmin})$	$\mathbb{P}(\text{Trust-threshold})$
$\varphi(\text{Malicious})$	$\mathbb{P}(\text{relay_id} \times \text{trust-val})$

received HELLO messages, a node compute the one-hop and one-hop neighbors.

$$\begin{aligned} \mathbf{R}(\text{Gen-Nlist}) = & \forall hp \in HP, \forall gn \in G - \text{Node} \mid \forall 1hl \in 1HL \\ & | 1hl[2] := \text{GenNeighbour}(hp, gn[1]) \wedge 1HL' = 1HL \cup \\ & \{(gn[1], 1hl[2])\} \wedge \forall 2hl \in 2HL | 2hl[2] := \text{Gen} - 2HN \\ & (hp, gn[1]) \wedge 2HL' = 2HL \cup \{(gn[1], 2hl[2])\} \end{aligned} \quad (\text{R1})$$

After populating one-hop and two-hop list, watchdog nodes and relay nodes are selected for monitoring and packet relaying purpose, respectively, as depicted in Algorithm 1. In rule R2, using the one-hop list, watchdog nodes are selected. The nodes that are not relay nodes and in the one-hop list of the relay node and source node are selected as watchdog. Also, the set of relay nodes are selected from one-hop neighbor list to reach two-hop neighbors. The transition *Gen-WDN* and *Gen-relay* is mapped to the following rules R2 and R3, respectively.

$$\begin{aligned} \mathbf{R}(\text{Gen-WDN}) = & \forall g1 \in G - 1HL, \forall mpl \in G - MPL, \forall wdl \\ & \in WDL, \forall gn \in Gn | gn[1] \notin mpl \longrightarrow wdl[1] := mpl[1] \wedge \\ & wdl[2] := mpl[2] \wedge wdl[3] := gn[1] \wedge WDL' = WDL \cup \\ & \{(wdl[1], wdl[2], wdl[3])\} \end{aligned} \quad (\text{R2})$$

$$\begin{aligned} \mathbf{R}(\text{Gen-relay}) = & \forall g1 \in G - 1HL, \forall g2 \in G - 2HL, \forall gn \in \\ & GN, \forall mpl \in \text{relay} - L \mid [\text{Con}(g1[1]_{i:g1[1]}, gn[1], g2[1]) > \\ & \text{Con}(g1[1]_{j:g1[1] \wedge j \neq i}, gn[1], g2[1]) \vee \text{Con} - \text{iso}(g1[1], gn \\ & [1], g2[1]) = \text{True}] \longrightarrow mpl[1] := gn[1] \wedge mpl[2] := g1 \\ & [1] \wedge \text{relay} - L' = \text{relay} - L \cup \{(mpl[1], mpl[2])\} \end{aligned} \quad (\text{R3})$$

So far, the watchdog and the relay nodes are selected. Now, the source node generates a message and wants to broadcast it into the network. In rule R4, the same process is depicted, where the source node generates the message and in response the watchdog nodes overhear it and the respective relay node receives it.

$$\begin{aligned} \mathbf{R}(\text{Broadcast}) = & \forall m \in \text{Msg}, \forall oh - sn \in OH - SN, \forall rm \in \\ & \text{Rec} - SN | oh - sn[1] := m[1] \wedge oh - sn[2] := m[2] \wedge oh - \\ & sn[4] := m[4] \wedge oh - sn[5] := m[5] \wedge OH - SN' = OH - \\ & SN \cup \{(oh - sn[1], oh - sn[2], oh - sn[3], oh - sn[4], \\ & oh - sn[5], oh - sn[6], oh - sn[7])\} \wedge rm[1] := m[1] \wedge \\ & rm[2] := m[2] \wedge rm[4] := m[4] \wedge rm[5] := m[5] \wedge \text{Rec} - \\ & SN' = \text{Rec} - SN \cup (rm[1], rm[2], rm[3], rm[4], rm[5], \\ & rm[6], rm[7], rm[8]) \end{aligned} \quad (\text{R4})$$

The relay node forwards the message that it received from the source node. When the relay node forwards the message, the watchdog nodes and the source node overhear the message forwarded by the relay node. The same is depicted in rule R5. We compute the trust of the relay nodes by (a) computing the number of messages forwarded by the relay nodes by analyzing the overheard messages of source node and watchdog nodes, (b) checking the contents of the message forwarded by the relay node, and (c) by investigating if the relay node generate its own control messages. The computations are performed the same way as explained in Algorithm 1.

$$\begin{aligned} \mathbf{R}(\text{Forward}) = & \forall ohm \in OH - \text{relay}, \forall rem \in \text{Get} - \text{Msg}, \forall f \\ & \in \text{Flood}, \forall ohsn \in OH - SNMP \mid rem[2] \neq \text{NULL} \wedge rem \\ & [8] = \text{Send}() \longrightarrow (f[1] := rem[1] \wedge f[2] := rem[2] \wedge f[3] \\ & := rem[4] \wedge f[4] := rem[5] \wedge \text{Flood} = \text{Flood} \cup (f[1], f[2], \\ & f[3], f[4]) \wedge (ohm[1] := rem[1] \wedge ohm[2] := rem[2] \wedge \\ & ohm[3] := rem[3] \wedge ohm[4] := rem[4] \wedge ohm[5] := rem[5] \\ & \wedge ohm[6] := rem[6] \wedge ohm[7] := rem[7] \wedge OH - \text{relay}' = \\ & OH - \text{relay} \cup (ohm[1], ohm[2], ohm[3], ohm[4], ohm[5], \\ & ohm[6], ohm[7]) \wedge ohsn[6] := rem[2] \wedge OH - SNMP = OH \\ & - SNMP \cup \{(ohsn[1], ohsn[2], ohsn[3], ohsn[4], ohsn[5], \\ & ohsn[6], ohsn[7])\} \end{aligned} \quad (\text{R5})$$

In rule R6, the source node computes the trust of relay node based on its own observations and those received from watchdog nodes according to Eq. 7. In rule R7, trust computed in rule R6 is compared to the trust threshold and if certain node trust falls below threshold, the node will be isolated from the routing path.

$$\begin{aligned}
R(\text{Trust-Obs}) = & \forall gsno \in Get - SNO, \forall gwdo \in Get - \\
& WDO, \forall t \in Trust \mid gsno[4] = gwdo[4] \wedge gsno[2] = gwdo[2] \\
& \longrightarrow t[1] (gsno[6]) \cup (gwdo[6]) gsno[2] \alpha \wedge Content(gsno \\
& [2], gsno[6]) = Content(gwdo[2], gwdo[6]) \longrightarrow t[2] \beta \wedge \\
& gsno[5] = TC \wedge gwdo[5] = TC \wedge Gen - TC - Pack(gwdo \\
& [4], gsno[4]) > gsno[7] \longrightarrow t[3] \gamma \wedge t[4], gsno[4] \wedge T' = \\
& T \cup \{(t[1], t[2], t[3], t[4])\}
\end{aligned}
\tag{R6}$$

$$\begin{aligned}
R(\text{Comp-Mali}) = & \forall gto \in Get - T, \forall gth \in Get - Th \forall \\
& cm \in Comp - Mali \mid Sum(gto[1], vgt[2], gto[3]) < gth \\
& \longrightarrow cm[1] := gto[4] \wedge cm[2] := Sum(gto[1], gto[2], gto[3]) \\
& CM' = CM \cup (cm[1], cm[2])
\end{aligned}
\tag{R7}$$

4.4 Verification of properties

In our analysis, we aim at verification of the following correctness properties.

- Property 1: common neighbors of source node S and relay node x having a trust greater than the trust threshold must be selected as watchdog nodes.
- Property 2: second-hand information must be considered from only those nodes which are designated as watchdog nodes.
- Property 3: second-hand information is considered from only those nodes having a trust value greater than the trust threshold and whose trust deviation is less than the deviation threshold.
- Property 4: A trust of a malicious node M misbehaving in the context of one of the trust attribute must be decremented as per the specification of MATF.

4.5 Verification results

To perform the verification of the HLPN models using Z3, we unroll the model M and the formula f (properties) that provides M_k and f_k , respectively. Moreover, the said formulas are then passed to Z3 to check if $M_k \models f_k$ (if the formula f holds in the model M up to the bound k execution time). The solver performs the verification and provide the results as satisfiable (*sat*) or unsatisfiable (*unsat*). If the answer is *sat*, then the solver will generate a counter example, which depicts the violation of the property or formula f . Moreover, if the answer is *unsat*, then formula or the property f holds in M up to the bound k (in our case k is exec. time). In these verification results, we verify the properties mentioned previously.

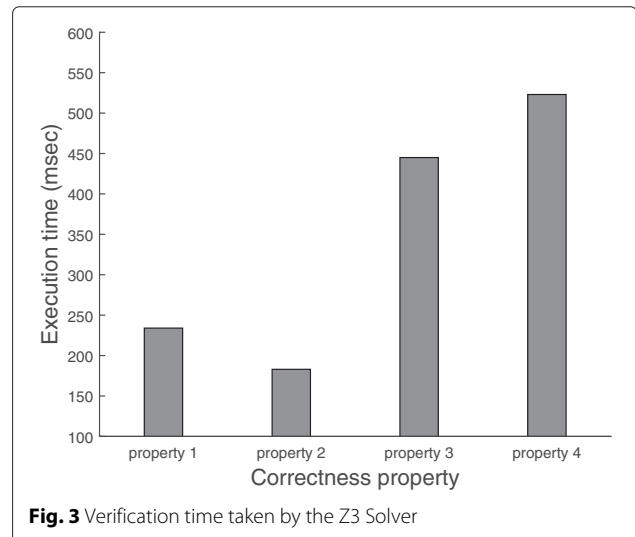
It is worth mentioning here that in our formal verification results, we verify the correctness properties of the proposed scheme, not the performance of the proposed scheme (for performance evaluation results, please refer to the Section 5).

Due to high time-consuming process, execution time is an important metric to verify the properties of the MATF. Figure 3 depicts the time taken by the Z3 Solver to prove that the properties discussed previously in Subsection 4.4 hold in the model.

5 Experimental performance analysis

In this section, we evaluate the performance of MATF in comparison to the scheme proposed in [14], referred to as single attribute-based trust framework (SATF) in what follows. Network simulator 2 (NS-2)[19, 39] is used to implement and analyze the performance of the proposed MATF. For the simulation experiments, we have varied the mobility speed of the nodes between 1 and 10 m/s. For data traffic, 30 % of the total nodes in the network are selected as source-destination pairs (sessions), spread randomly over the network. Only 512-byte data packets are sent. The packet sending rates in each pair are varied to change the offered load in the network. All traffic sessions are established at random times near the beginning of the simulation run and stay active until the end. Moreover, a very popular and commonly used mobility model, called random way point mobility model [40], is used for node mobility. In the aforementioned mobility model, each node selects a random destination and starts moving with a randomly chosen speed (uniformly distributed between 0 and a predefined maximum speed).

The trust threshold value is 0.4 in this set of experiments [14], which is the maximum tolerated misbehavior for a node to be a part of the network [41]. A trust threshold



value determines the trust level that a node has to maintain to be a legitimate node. To handle high-dimensional parameter space, we define some commonly used simulation parameters, as stated in Table 4. The number of simulation experiments has been chosen sufficiently large in order to get 95 % confidence interval for the results.

5.1 Experimental adversarial model

In our adversarial model, the malicious node count is set to 10–30 % of the total nodes in the network. In order to evaluate the proposed scheme against the adversary nodes thoroughly, malicious nodes are selected randomly to keep their distribution uniform in the network. In our experiments, we simulated packet dropping attack by having malicious nodes dropping control and data packets randomly or selectively with 25 % probability. Moreover, malicious nodes are also misbehaving by launching the withholding attack against the legitimate nodes. In withholding attack, misbehaving node does not generate control traffic as per the specification of the routing protocol. Because of the aforementioned behavior of misbehaving nodes, legitimate nodes are unable to have a consistent and updated view of the network. Furthermore, number of malicious nodes exercise bad-mouthing and false praise attacks in collusion is varied from 10 to 50 % of the total nodes in the simulation scenarios.

5.2 Simulation results and analysis

We now discuss the results of the comparison between the MATF and the SATF in terms of several performance metrics.

5.2.1 Impact of trust deviation threshold

Trust deviation threshold means that second-hand information whose deviation from an evaluating node's observations is greater than the aforementioned threshold will be filtered out while computing the evaluated node trustworthiness. To select the best optimal trust deviation

threshold to filter second-hand information, we simulate the MATF for varying the deviation threshold with increasing number of dishonest nodes. For this set of simulation, the mobility speed is set to 1–4 m/s. Dishonest nodes exercise the false praise and bad-mouthing attacks to show the impact on detection rate and false positives rate, respectively (Fig. 4).

Figure 5a, b illustrates the impact of increasing number of dishonest nodes on the false positive rate and detection rate under different trust deviation thresholds. It can be inferred from Fig. 5a that detection rate is first increasing up to the deviation threshold of 0.4 and then decreasing with increasing number of dishonest nodes. The reason is that with higher trust deviation threshold, false recommendations from bad-mouthing nodes are not filtered out during the trust computation of evaluated nodes, which provides more opportunities to misbehaving nodes to remain undetected.

Similarly, Fig. 5b shows the impact of varying trust deviation threshold for increasing number of dishonest nodes. It is obvious from the figure that with increasing trust deviation threshold, the false positives rate is also increasing. The reason is that with higher deviation threshold, such as 0.5 and 0.6, false recommendation from bad-mouthing nodes having deviation of 60 % are only filtered out which causes legitimate nodes as misbehaving nodes, hence more false positives rate.

It can be summarized from the above results that 0.4 is an optimal trust deviation threshold in terms of detection rate and false positives. It is worth mentioning here that we will use the trust deviation threshold of 0.4 for the rest of the simulation scenarios.

5.2.2 Trust values

Figure 4 shows the trust values computation of a some specific misbehaving node at different simulation time instances. As shown in the figure, the MATF decrements

Table 4 Simulation parameters

Simulation time	1000 s
Number of nodes	60
Number of malicious nodes	10–30%
Network size	1000 × 1000 m
Transmission range	250 m
Max speed	1–10 m/s
Mobility model	Random way point
Traffic type	Constant bit rate (CBR)
Trust threshold	0.4
Deviation threshold	0.4
Energy model	Generic energy model [44]

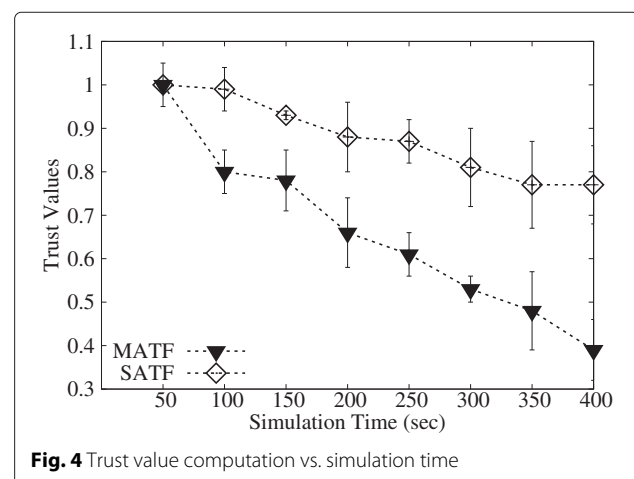


Fig. 4 Trust value computation vs. simulation time

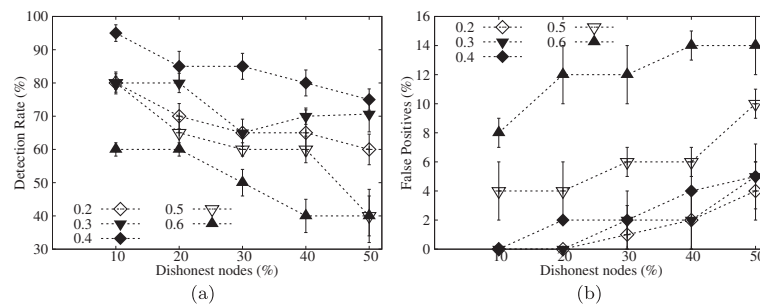


Fig. 5 Impact of trust deviation threshold on detection rate and false positive rate (a, b)

the trust in an expedite way of the misbehaving node to achieve the threshold because of multi-attribute and efficient dishonest recommendation filtration criteria, hence more informed decisions. The MATF evaluates the evaluated node on the basis of different network functions, hence more informed and prompt decisions about the trustworthiness of nodes can be taken. However, in case of the SATF, the trust is computed slowly due to high bootstrapping time and data sparsity problem. The reason for this behavior is that evaluated nodes are observed in the context of data forwarding only. It can be inferred from Fig. 4 that the MATF efficiently overcomes the bootstrapping and data sparsity at the start-up of the network as compared to SATF.

5.2.3 Detection time and detection rate

Detection time refers to the time taken by the trust-based security scheme to detect and declare a misbehaving node as a malicious node. Similarly, malicious node detection rate is calculated as the percentage of malicious nodes detected among the total number of malicious nodes within the network.

Figure 6a shows the malicious node detection time for increasing node speed in the MATF and the SATF. Aforementioned figure shows that the time required in case of the MATF for increasing node speed is smaller as compared to the SATF. The detection time required for misbehaving node detection in the SATF is almost double the MATF. The reason for this behavior is the slow trust building process as discussed in the Fig. 4 analysis. Overall, the detection time is increasing for increasing node speed. This is because of the fact that for higher node speed, nodes have smaller time of interaction; hence, it takes time to build the trust under the high node mobility.

Figure 6b shows the detection rate for increasing node speed. As shown in figure, detection rate is higher in case of the MATF. The reason is that in the MATF, the node's trust is analyzed in multiple contexts, which expedite the detection rate. Similarly, Fig. 6c shows the malicious node detection rate with the simulation time. The figure shows that the percentage of the malicious node detection is

higher in case of the MATF as compared to the SATF. The detection rate is 100 % at time $t = 500$ s in the MATF, while half of the malicious nodes are detected in the case of the SATF.

Figure 6d illustrates the impact of increasing the number of nodes on the detection rate while keeping the mobility fixed at 1–6 m/s. It can be inferred from the figure that there is a slight increase in the detection rate with increasing node density. This is due to the fact that under high node density, higher number of watchdogs will be available to observe the behavior of an evaluated node that leads to better detection rate.

The impact of colluding dishonest attackers on detection rate is shown in Fig. 6e. As the figure shows, MATF scheme is able to keep the detection rate nearly about 90 % even in case of higher number of false praise nodes as compared to SATF. The reason is the implementation of an efficient trust deviation criteria, hence more confident decisions. Due to efficient trust deviation criteria, recommendations from colluding dishonest attackers are filtered out and are not considered in the trust computation of an evaluated node.

5.2.4 False positive rate

The false positive rate is the ratio of the legitimate nodes declared as malicious to the total number of legitimate nodes.

Effect of node speed on false positive rate is shown in Fig. 7a, under the MATF and the SATF. Figure 7a illustrates that false positive rate is much lower in the MATF as compared to the SATF. The reason for the aforementioned behavior is that MATF uses the second-hand information from only designated nodes which have a deviation in trust values less than the deviation threshold, hence more informed decisions about the node's trustworthiness. While in case of the SATF, second-hand information are used from all the neighbor nodes to compute the trustworthiness of a node. As there are some nodes deployed in the network, exercising the bad-mouthing attack against the legitimate nodes causes higher false positives rate in the SATF. Overall, the figure shows that with an increase

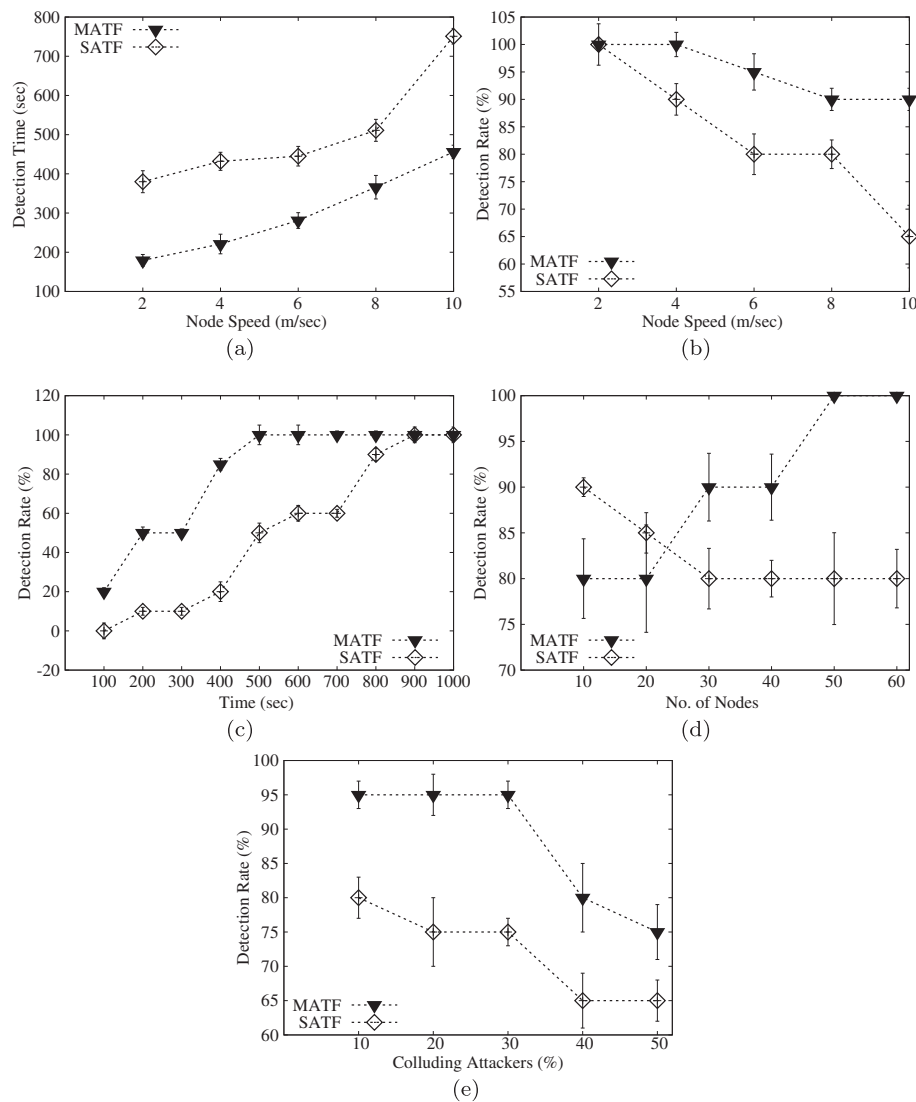


Fig. 6 Effect on detection rate (a–e)

in the node speed, the false positives rate also increases. The aforementioned behavior is due to the fact that an evaluating node and the watchdog nodes cannot differentiate between intentional and unintentional malicious activities of a node. For example, even if a node fails to forward a packet because of the network conditions, it is regarded as a malicious activity by a node. As a result, under high node speed, the false positives rate increases.

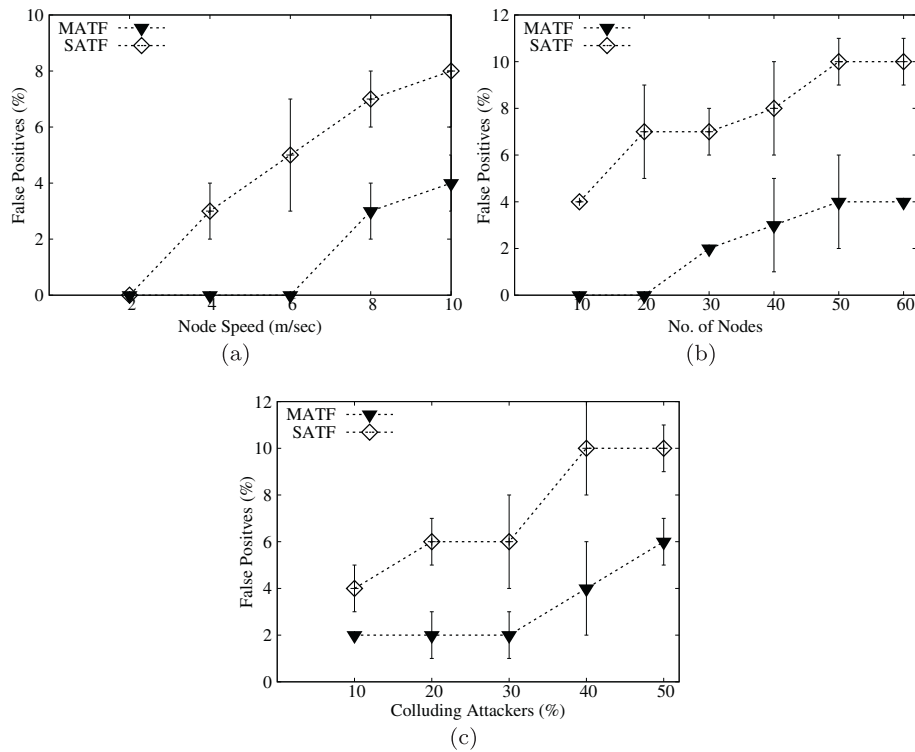
Similarly, Fig. 7b shows the effect of increasing node density on false positive rate. The figure illustrates that for increasing node density, the false positive rate in case of the MATF is lower as compared to the SATF. The reason is that more legitimate nodes are selected as watchdog, which provides accurate and precise information about the trustworthiness of the evaluated nodes and also because of using an efficient filtration criteria to filter the

dishonest recommendations. In case of the SATF, the false positive rate is increasing as the number of bad-mouthing and false praising nodes are also increasing, which causes a false trust estimation about the legitimate nodes.

Figure 7c shows the impact of dishonest colluding attackers on false positive rate. It is obvious from the figure that MATF withstands effectively against the increasing dishonest nodes in terms of false positives. The reason is the use of an efficient trust deviation criteria in the proposed scheme as previously discussed in the reasoning of Fig. 6e.

5.2.5 Packet delivery ratio

Packet delivery ratio (PDR) is the ratio of the number of data packets generated by a source node and the number of packets received at the destination. With malicious

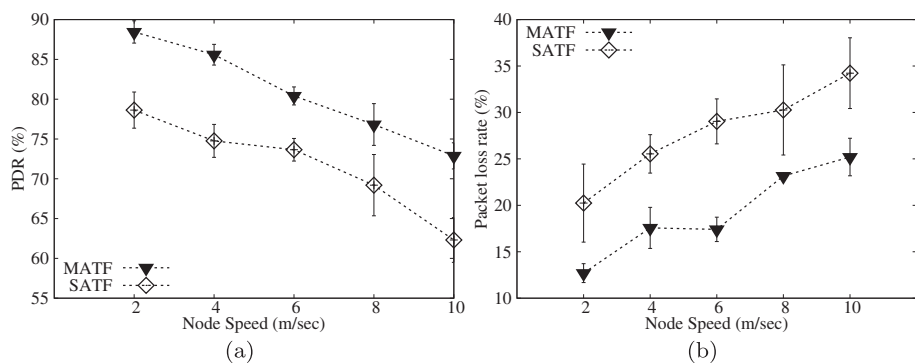
**Fig. 7** Effect on false positives (a–c)

node count set to 20 % of the total number of deployed nodes, the control and data packet dropping and withholding attacks are implemented. Figure 8a illustrates the effect of the mobility speed of the nodes on the PDR while keeping the data rate constant at 4 kbps. Figure 8a shows that the MATF has higher PDR as compared to the SATF as it isolates malicious nodes from the routing paths very earlier (as shown in Fig. 6c). Moreover, it can also be observed that the PDR decreases with increasing node speed. The reason for the aforementioned behavior is that at a higher node speed, the node drops packets due to the frequent link changes. These results illustrate that the

MATF eliminates the malicious nodes well in time from the network and improves the PDR by 10–12 % for varying mobility speeds of the nodes.

5.2.6 Packet loss rate

In this section, we present the packet loss analysis of the proposed MATF. Although the packet delivery ratio provides the big picture of efficiency and effectiveness of any scheme, however, the reason to present the packet loss analysis in this paper is to show the effectiveness of the MATF scheme in terms of reducing the packet loss due to misbehaving nodes. As there are many reasons of packet

**Fig. 8** Effect on PDR and packet loss rate (a, b)

loss in MANETs, such as packet loss due to link errors, queue overflow, frequent link changes, and malicious drop [42, 43]. In these simulation results, we consider the packet loss that is only caused by the malicious node-dropping packets. Figure 8b shows the packet loss rate for the increasing node speed in the MATF and the SATF. The results show that the MATF has about 8–15 % less packet loss rate as compared to the SATF. The reason for this behavior is that misbehaving nodes are detected and isolated well in time on the basis of multi-attribute trust criteria. However, in case of the SATF, the misbehaving nodes are detected and isolated very late in the simulation (as shown in Fig. 6c), which provides more packet drop opportunities to the misbehaving nodes.

5.2.7 Energy consumption

The major causes of the energy consumption in MANETs are the packet transmission and reception. To compute the energy consumed by the nodes in both the MATF and SATF schemes, we use the generic energy model supported by NS-2. The generic energy model can estimate the consumption of energy for continuous and variable transmission power levels. The parameters we used are as follows: 100 J of initial energy, 0.05 W for transmission, 0.02 W for reception, 0.01 W for idling, and 0.0 W when sleeping. It is worth mentioning that energy consumed is shown in percentage in these results, which is the total percentage energy consumption of the initial energy of a node. The energy consumption of the proposed MATF in comparison to the SATF is shown in Fig. 9a. As there is no extra message communication in the MATF in comparison to the SATF, the figure shows that energy consumption is almost equal to that of the SATF. A slight increase in the energy consumption in case of MATF is because of the nodes in MATF requiring some extra processing to compute the trust of the nodes on the basis of multi-attribute trust criteria. Moreover, the packet delivery ratio is higher and packet loss due to malicious nodes is lower in the MATF in comparison to the SATF,

which also causes more energy consumption as packets need to travel more longer paths in the network, hence more energy consumption at those nodes in the routing path.

5.2.8 Normalized routing load

Normalized routing load (NRL) is the ratio of the total number of control packets transmitted by the nodes to the total number of received data packets at the destination nodes. It is used to evaluate the efficiency of a routing protocol.

Figure 9b illustrates that NRL is smaller in the MATF as compared to the SATF. The reason is the more packet delivery ratio per control packets in the MATF. As the SATF suffers from more packet loss as shown in the figure, control packets sent per data packet is higher, which causes higher NRL in the SATF. Overall, the routing overhead is increasing in both the schemes with an increase in the node speed. The reason for this behavior is that to maintain the routes under high node mobility, more control packets are transmitted.

6 Security analysis

In this section, we present the security analysis of the proposed MATF against the various attacks.

6.1 Security against bad-mouthing and false praise attack

In the MATF, second-hand information is considered from only those nodes, which are designated as watchdog nodes, having trust value greater than the trust threshold, and trust deviation is less than the deviation threshold. Due to the aforementioned criteria for second-hand information, the MATF effectively withstands against the bad-mouthing and false praise attacks.

6.2 Security against selective misbehavior

A smart adversary node may misbehave selectively, such as drops data packets, while forwards control packets. Depending upon the security requirements and the

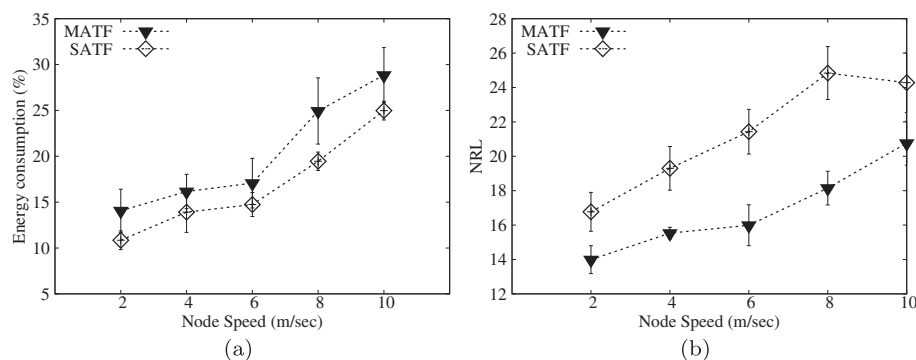


Fig. 9 Effect on energy consumption and NRL (a, b)

privilege provided by the MATE, an evaluating node can selectively use smart misbehaving nodes to perform different network functions. For example, if an adversary node misbehaves by dropping data packets only, then an evaluating node can use such a node for other network functions, such as control packet forwarding.

6.3 Security against colluding attackers

In the proposed scheme, an evaluating node uses the trust attributes based on local states and its own observation; collusion attack is not much effective against the scheme. The only collusion attack that is possible against the scheme is the publication of false-praise and bad-mouthing information against the legitimate nodes. In the proposed MATE, efficient trust deviation criteria are used which filter such false-praise and bad-mouthing information, as discussed in Figs. 6e and 7c. Results presented in the aforementioned figures reveal that the proposed MATE scheme efficiently withstands against the colluding attackers up to 30 % of the total nodes.

7 Conclusion and future work

In this work, we proposed a scheme that is based on the multi-attribute trust criteria to minimize the bootstrapping time and to deal with the selective misbehavior. The proposed trust model augments the security of a MANET by enabling a node to identify and remove malicious nodes from the routing paths by overhearing transmission at multiple nodes (evaluating node and watchdog nodes). The proposed security scheme not only provides a way to detect attacks and malicious behavior accurately and timely but also reduces the number of false positives by using the concept of multi-watchdogs. The proposed trust model is evaluated in the context of OLSR routing protocol. Moreover, to prove the correctness of the proposed scheme, we also presented a formal verification of our proposed MATE using HLPN, SMT-Lib, and Z3 Solver. Comparison between the MATE and the SATF has shown that our proposed scheme has more efficiently detected malicious nodes. Moreover, the MATE has shown promising results under high mobility speed of the nodes and frequent topology changes.

Simulation results show that the proposed trust model achieves 98–100 % detection rate of malicious nodes with only 1–2 % false positives. The proposed MATE has an improved packet delivery ratio in comparison to the SATF of about 90–75 and 80–65 %, respectively, in a network with malicious nodes.

We plan to extend our work by using the adaptive mechanism for the weight assignment to different trust attributes based on the run-time network conditions. Moreover, we will evaluate our proposed scheme as an

extension to some other reactive routing protocol like DSR to analyze the effect of underlying routing protocol.

Endnote

¹The difference between the trust values of a recommender node and an evaluating node about a particular evaluated node.

Acknowledgments

The work reported in this paper has been partially supported by the Higher Education Commission (HEC), Pakistan.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Department of Computer Sciences, COMSATS Institute of Information Technology, Islamabad, Pakistan. ²Department of Computer Sciences, COMSATS Institute of Information Technology, Abbottabad, Pakistan.

Received: 20 April 2016 Accepted: 9 August 2016

Published online: 24 August 2016

References

1. S Zhao, A Aggarwal, S Liu, H Wu, in *IEEE Wireless Communications and Networking Conference (WCNC2008)*. A secure routing protocol in proactive security approach for mobile ad-hoc networks (IEEE, Las Vegas, 2008), pp. 2627–2632. 10.1109/WCNC.2008.461
2. YC Hu, A Perrig, DB Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wirel. Netw.* **11**, 21–38 (2005)
3. P Papadimitratos, ZJ Haas, in *IEEE Applications and the Internet Workshops*. Secure link state routing for mobile ad hoc networks (IEEE, Orlando, 2003), pp. 379–383
4. MS Obaidat, I Woungang, SK Dhurandher, V Koo, *A cryptography-based protocol against packet dropping and message tampering attacks on mobile ad hoc networks security and communication networks*. (John Wiley & Sons, Ltd, Malden MA, 2014), pp. 376–384
5. T Zahariadis, P Trakadas, HC Leligou, S Maniatis, P Karkazis, A novel trust-aware geographical routing scheme for wireless sensor networks. *Wirel. Pers. Commun.* **69**(2), 805–826 (2013)
6. G Zhan, W Shi, J Deng, Design and implementation of TARF: a Trust-Aware Routing Framework for WSNs. *IEEE Trans. Dependable Secure Comput.* **9**(2), 184–197 (2012)
7. S Buchegger, JY Le Boudec, in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. Performance analysis of the CONFIDANT protocol (ACM, New York, 2002), pp. 226–236
8. A Chakrabarti, V Parekh, A Ruia, in *Advances in Computer Science and Information Technology: Networks and Communications (Springer)*. A trust based routing scheme for wireless sensor networks (Springer, Berlin Heidelberg, 2012), pp. 159–169
9. P Michiardi, R Molva, in *Advanced communications and multimedia security*. Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks (Springer, USA, 2002), pp. 107–121
10. S Ganerwal, LK Balzano, MB Srivastava, Reputation-based framework for high integrity sensor networks. *ACM Trans. Sens. Netw. (TOSN)*. **4**(3), 15 (2008)
11. O Khalid, SU Khan, SA Madani, K Hayat, MI Khan, N MinAllah, J Kolodziej, L Wang, S Zeadally, D Chen, Comparative study of trust and reputation systems for wireless sensor networks. *Secur. Commun. Netw.* **6**(6), 669–688 (2013)
12. A Ahmed A, KA Bakar, MI Channa, K Haseeb, AW Khan, A survey on trust based detection and isolation of malicious nodes in ad hoc and sensor networks. *Front. Comput. Sci.* **9**(2), 280–296 (2015)
13. S Marti, TJ Giuli, K Lai, M Baker, in *ACM Proceedings of the 6th annual international conference on Mobile computing and networking*. Mitigating routing misbehavior in mobile ad hoc networks (ACM, New York, 2000), pp. 255–265
14. AM Shabut, KP Dahal, SK Bista, IU Awan, Recommendation based trust model with an effective defence scheme for MANETs. *IEEE Trans. Mob. Comput.* **14**(10), 2101–2115 (2015)

15. FS Proto, A Detti, C Pisa, G Bianchi, in *IEEE International Conference on Communications (ICC)*. A framework for packet-droppers mitigation in OLSR wireless community networks (IEEE, Kyoto, 2011), pp. 1–6
16. JM Robert, H Otrok, A Chriqi, RBC-OLSR: Reputation-based clustering OLSR protocol for wireless ad hoc networks. *Comput. Commun.* **35**(4), 487–499 (2012)
17. D Zhang, CK Yeo, Distributed court system for intrusion detection in mobile ad hoc networks. *Comput. Secur.* **30**(8), 555–570 (2011)
18. SU Malik, SU Khan, Formal methods in LARGE-SCALE computing systems. *ITNOW*. **55**(2), 52–53 (2013)
19. T Issariyakul, E Hossain, *Introduction to network simulator NS2*. (Springer Science & Business Media, USA, 2011)
20. S Tan, X Li, Q Dong, Trust based routing mechanism for securing OSLR-based MANET. *Ad Hoc Netw.* **30**, 84–98 (2015)
21. EM Shakshuki, N Kang, TR Sheltami, EAACK—a secure intrusion-detection system for MANETs. *IEEE Trans. Ind. Electron.* **60**(3), 1089–1098 (2013)
22. K Liu, J Deng, PK Varshney, K Balakrishnan, An acknowledgment-based approach for the detection of routing misbehavior in MANETs. *IEEE Trans. Mob. Comput.* **6**(5), 536–550 (2007)
23. TR Sheltami, A Basabaa, EM Shakshuki, A3ACKs: adaptive three acknowledgments intrusion detection system for MANETs. *J. Ambient Intell. Humanized Comput.* **5**(4), 611–620 (2014)
24. P Gallagher, C Furlani, Digital signature standard (DSS). Federal Information Processing Standards Publications, volume FIPS (2013), 186–3 (2013)
25. RL Rivest, A Shamir, L Adleman, A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*. **26**(1), 96–99 (1983)
26. S Bansal, M Baker, Observation-based cooperation enforcement in ad hoc networks. *Research Report cs.NI/0307012*, Stanford University, 120–130 (2003)
27. X Li, Z Jia, P Zhang, R Zhang, H Wang, Trust-based on-demand multipath routing in mobile ad hoc networks. *IET Inf. Secur.* **4**(4), 212–232 (2010)
28. H Xia, Z Jia, X Li, L Ju, EH Sha, Trust prediction and trust-based source routing in mobile ad hoc networks. *Ad Hoc Netw.* **11**(7), 2096–2114 (2013)
29. A Adnane, C Bidan, RT de Sousa, Trust-based security for the OLSR routing protocol. *Comput. Commun.* **36**(10), 1159–1171 (2013)
30. A Adnane, in *Proceedings of the 2008 ACM symposium on Applied computing*. Autonomic trust reasoning enables misbehavior detection in OLSR (ACM, New York, 2008), pp. 2006–2013
31. D Kukreja, SK Dhurandher, BVR Reddy, *Enhancing the Security of Dynamic Source Routing Protocol Using Energy Aware and Distributed Trust Mechanism in MANETs. Intelligent Distributed Computing*. (Springer International Publishing, Springer Switzerland, 2015), pp. 83–94
32. R Abdellaoui, J Robert, in *4th Conference on Security in Network Architectures and Information Systems (SAR-SSI)*. Su-olsr: A new solution to thwart attacks against the olsr protocol, (Luchon, 2009), pp. 239–245
33. D Câmara, AA Loureiro, F Filali, in *IEEE Global Telecommunications Conference (GLOBECOM'07)*. Methodology for formal verification of routing protocols for ad hoc wireless networks (IEEE, Washington, 2007), pp. 705–709
34. SU Malik, SU Khan, SK Srinivasan, Modeling and analysis of state-of-the-art VM-based cloud management platforms. *IEEE Trans. Cloud Comput.* **1**(1), 1–1 (2013)
35. F Ghassemi, S Ahmadi, W Fokkink, A Movaghar, *Model checking MANETs with arbitrary mobility*, (2013), pp. 217–232
36. C C Barrett, A Stump, C Tinelli, *The Satisfiability Modulo Theories Library (SMT-LIB)*, (2010). <http://smtlib.cs.uiowa.edu/>. Accessed 15 Jan 2016
37. L De Moura, N Bjørner, in *Tools and Algorithms for the Construction and Analysis of Systems. Z3: An efficient SMT solver* (Springer, Berlin Heidelberg, 2008), pp. 337–340
38. SU Malik, SK Srinivasan, SU Khan, L Wang, in *12th International Conference on Scalable Computing and Communications (ScalCom)*. A methodology for OSPF routing protocol verification (IEEE, Changzhou, 2012)
39. P Whigham, *The VINT project, the network simulator - ns-2*. (University of Otago, 2003), pp. 181–186. <http://www.isi.edu/nsnam/ns/>. Accessed 05 Jan 2016
40. J Broch, DA Maltz, DB Johnson, YC Hu, J Jetcheva, in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*. A performance comparison of multi-hop wireless ad hoc network routing protocols (ACM, New York, 1998), pp. 85–97
41. MS Khan, D Midi, MI Khan, E Bertino, in *IEEE Trustcom/BigDataSE/ISPA, Vol.1*. Adaptive trust threshold strategy for misbehaving node detection and isolation (IEEE, Helsinki, 2015), pp. 718–725
42. Z Wei, H Tang, FR Yu, P Mason, in *IEEE Military Communications Conference (MILCOM)*. Trust establishment based on Bayesian networks for threat mitigation in mobile ad hoc networks (IEEE, Perundurai, 2014), pp. 171–177
43. Y Lu, Y Zhong, B Bhargava, *Packet Loss in Mobile Ad Hoc Networks*. (IEEE, Baltimore, 2003)
44. Technical Report CSD-TR 03-009. Department of Computer Science, Purdue University (2003). <http://docs.lib.purdue.edu/cstech/1558/>. Retrieved 26 Dec 2015.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com