Computing (2019) 101:791–818
https://doi.org/10.1007/s00607-018-0685-7

A fuzzy security protocol for trust management in the internet of things (Fuzzy-IoT)

Mohammad Dahman Alshehri[1,2] · **Farookh Khadeer Hussain**[1]

Received: 28 April 2018 / Accepted: 22 November 2018 / Published online: 10 December 2018

Abstract

Recently, the Internet of things (IoT) has received a lot of attention from both industry and academia. A reliable and secure IoT connection and communication is essential for the proper working of the IoT network as a whole. One of the ways to achieve robust security in an IoT network is to enable and build trusted communication among the things (nodes). In this area, the existing IoT literature faces many critical issues, such as the lack of intelligent cluster-based trust approaches for IoT networks and the detection of attacks on the IoT trust system from malicious nodes, such as bad service providers. The existing literature either does not address these issues or only addresses them partially. Our proposed solution can firstly detect on-off attacks using the proposed fuzzy-logic based approach, and it can detect contradictory behaviour attacks and other malicious nodes. Secondly, we develop a fuzzy logic-based approach to detect malicious nodes involved in bad service provisioning. Finally, to maintain the security of the IoT network, we develop a secure messaging system that enables secure communication between nodes. This messaging system uses hexadecimal values with a structure similar to serial communication. We carried out extensive experimentation under varying network sizes to validate the working of our proposed solution and also to test the efficiency of the proposed methods in relation to various types of malicious behavior. The experiment results demonstrate the effectiveness of our approach under various conditions.

✉ Mohammad Dahman Alshehri
Mohammad.Alshehri@uts.edu.au

Farookh Khadeer Hussain
Farookh.Hussain@uts.edu.au

[1] Centre for Artificial Intelligence, School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia

[2] Computer Science Department, Computers and Information Technology College, Taif University, Taif, Kingdom of Saudi Arabia

Keywords Internet of things (IoT) · Trust management · Cyber security · Protocol · Fuzzy logic

Mathematics Subject Classification 68 Computer Science

1 Introduction

The Internet of things (IoT) is a vibrant new field of research in electronic engineering and computer networks. It has transformed the Internet from interaction between humans only to that between humans and things and even interaction between things [22]. This has been made possible through the development of smart devices which are able to make decisions without the intervention of humans and share information with other smart devices to achieve a particular goal. However, the incorporation of all these devices into the standard Internet leads to various challenges in security since the majority of Internet technologies and communication protocols were not originally designed for IoT support [13]. The distributed and decentralized nature of IoT is also a challenge in terms of access control, trust management and identity management (IdM) [19]. Trust management is the most important in a network of heterogeneous objects such as IoT [9]. In order to address this issue of control, an existing solution is standardisation through an IoT gateway [30]. Unfortunately, this requires extra infrastructure. However, the fuzzy approach to trust-based access control (FTBAC) can be used for trust-based dynamic access control. FTBAC is a scalable and flexible framework where an increased number of devices has no effect on their performance and functionality. This approach to trust management achieves cryptographic protection through access control by increasing levels of trust even though this creates extra overhead due to energy and time consumption. The model is easily integrated in decision-making based on utility as its flexibility allows for additional components [19]. However, there are several limitations to this approach and also in the existing work in the IoT area in terms of demonstrating the dynamics of moving nodes in an IoT network and how communication between IoT nodes can be both sustainable and secure. The trust management protocol, scalability and context are all important factors of IoT [4]. In order to address these issues, we propose a new approach of fuzzy security protocol and trust management for IoT-based clusters by developing a secure method of communication and exchanging messages between IoT nodes using a novel security protocol for the IoT. This protocol allows nodes to move from one cluster to another in a secure way. Further, our proposed protocol uses a message system similar to serial communication for secure message encryption. Moreover, we use fuzzy logic in our approach to detect malicious nodes and to restrict their untrusted function of speaking incorrect recommendations about the nodes in the network. Further, the effectiveness of fuzzy logic in the detection of bad service providers, on-off attacks and contradictory behaviour attacks (con-behaviour) is demonstrated.

The remainder of this paper is organized as follows: Sect. 2 reviews the related work in literature. Section 3 details a secure message system between IoT nodes. Section 4 overviews the trust and fuzzy algorithms. Section 5 presents an intelligent security protocol. Section 6 discusses the simulations, results and analysis. The paper is concluded in Sect. 7.

2 Related work

Mosenia and Jha [22] found that the IoT paradigm led to the development of several protocols of communication with the miniaturization of transceivers which provides the opportunity to transform an isolated device. Advances in technology have exponentially increased the number of Internet-connected computing and sensing devices. However, these suffer from the possibility of attacks and potential threats to privacy and security, especially when data is transferred from one cluster to another. A variety of approaches has been proposed to address the problem of IoT trust management, even though it has been argued and acknowledged that the real challenge for trust management is scalability. Therefore, there is a need to consider the development of intelligent next-generation methods of trust management for IoT networks which accommodate the leaving and joining of nodes and handle large-scale networks [5].

To address this problem, Alshehri and Hussain [3] suggest a novel centralized trust management mechanism for IoT (CTM-IoT) on the basis of the super node (SN) in proposing the trust management mechanism for the Internet of things (TM-IoT), designed for the provision of trustworthy communication between nodes. The IoT system evolution involves the creation of nodes which require the trust management protocol to permit the establishment of an accurate trust network. This requires a dynamic system designed to deal with the threat of socially uncooperative and malicious nodes. Therefore, their proposal is based on achieving trustworthy communication between nodes through the division of the IoT environment into clusters. For each cluster, there is a master node (MN) as the local trust manager.

The use of radio frequency identification (RFID) tags through the Internet enables the identification of tags through the use of an appropriate authentication protocol [16]. The proposition is to use an encryption method on the basis of XOR manipulation for privacy protection and anti-counterfeiting. The logic in this design is based on service-oriented architecture targeting the relation between unique identifiers with particular services. Ray et al. [24] proposed an RFID security protocol and framework based on the customizability and scalability of the issues to support IoT implementation. They proposed an identification technique for a group-based and collaborative approach (hybrid approach) and security check handoff.

A clustering-driven intelligent trust management methodology (CITM-IoT) was suggested by Alshehri et al. [5] where IoT nodes are grouped into clusters based on their trust value. The IoT nodes in a cluster are able to progressively loss or gain trust values during their interaction with other nodes. Using a scalable trust management solution (IoT-TM) provides a trustworthy communication platform between the devices communicating with other nodes in the IoT system. The architecture of IoT-TM allows for heterogeneous IoT applications and devices to contact each other in trusted heterogenic communication between devices. The centralized model involves a master node and several clusters, allowing for the centralized trust management of things over a network. This creates a distributed trust system for CNs to communicate with each other, and for MNs to communicate cooperatively with the SN and the CNs in their cluster. In this framework, the achievement of scalability is based on placing the IoT nodes into clusters or groups based on their trust values.

Kotis et al. [15] focus on the trustworthiness of an entity, noting that in a distributed or open IoT environment, there are multiple generic applications and third-party devices that need to be securely deployed. Their suggestion is that the semantic interoperability approaches related to IoT need to be extended through trust semantics. In IoT, semantics refers to the ability to extract knowledge using various machines for the required services to be provided [2]. Trust semantics are used in describing the trust-related and quality attributes for the sources and their providers. Since the high heterogeneity level in IoT can magnify security threats during interactions, it is important to semantically enable trust in the open and distributed IoT to secure and ensure the deployment and selection of heterogeneous IoT entities without central authorities of trust.

Ahmed et al. [1] designed a trust and energy-aware routing protocol (TERP) to address the challenges of trust-based routing protocols. The TERP design is centered on energy efficiency and trustworthiness with the capability of the dynamic detection and isolation of misbehaving nodes during the phase of trust evaluation with the incorporation of an energy awareness feature in the route setup phase of the routing protocol. This helps in the better balancing of load among trusting nodes. The design also integrates trust-based routing with the additional inclusion of mechanisms to ensure the selection of end-to-end routes with the current energy levels of intermediate nodes. The evaluation of TERP, based on simulation in NS-2 indicates better performance regarding average energy consumption, throughput and lifetime of the network.

A trust-based secure routing protocol was proposed by Renubala and Dhanalakshmi [26] utilizing fuzzy-log in trust-based secure routing. The method utilizes bio-inspired energy-efficient cluster (BEE-C) protocol characteristics taking into consideration distance, battery level and node density. It also detects the black region on the network and enhances network security.

Chen et al. [10] proposed a trust and reputation model to defend large distributions of sensor networks in IoT/CPS against malicious attacks on nodes, especially because mechanisms of trust establishment can stimulate collaboration between the IoT nodes. Their approach facilitates the detection of untrustworthy entities and assists in the process of decision-making for various communication protocols. The focus is on a fuzzy theory-based trust and reputation model for the IoT/CPS environment which analyzes the unique and special features of trust challenges, the concept of trust and reputation, trust evaluation metrics, global trust relationship evaluation and local trust relationship evaluation. The fuzzy-based secure routing approach effectively protects WSNs from severe attacks through the dynamic replaying of routing information.

Lize et al. [18] developed a trust mechanism for IoT, establishing a formal trust management control mechanism based on the modeling architecture of IoT. They adopted a formal semantic-based method and fuzzy set theory in the realization of the mechanism of trust and decision-making based on trust for a reasonable and coherent result. The focus is on the decomposition of IoT into three layers, each under the control of trust management for special purposes. The process indicates that the final decision-making is done by a service requester and then uses a formal semantics-based and fuzzy set theory.

Sirisala and Bindu [27] proposed the uncertain rule-based fuzzy logic QoS trust model in MANETs (FQTM), selecting those nodes with cooperativeness and capability. Fuzzy logic was applied to compute the trust values of nodes in consideration of reliability and quality metrics. FQTM selects nodes with high trust values to construct routes to the destination. The expert system in the fuzzification process converts the crisp values using a rule base into fuzzy values where all the rules are framed according to the resource status of the nodes.

By investigating machine-to-machine (M2M) networks, Tuna et al. [28] examined security requirements such as resilience and availability against external entity attacks, increasing privacy and anonymity of the devices. Their proposition is based on using a technique derived from information control theory, tagging data, and providing various properties of privacy. However, they suggest the need for an overall integrated security approach to ensure that in M2M applications, there is end-to-end security.

Lin et al. [17] focused on fog/edge computing so that devices with computing services can be deployed at the network edge with the aim of improving the experience of users and the resilience of the services when failures occur. With the advantage of being close to end-users and distributed architecture, the approach provides greater quality of service for IoT applications and a faster response. This makes it suitable for future IoT infrastructure to cover the privacy and security issues in the intelligent cyber world.

In looking at the security of IoT frameworks, Ammar et al. [7] considered eight frameworks based on their architecture, compatible hardware, the essentials of third-party smart app development and security features. The comparison showed that similar standards are used in communication security while different methodologies are used in the provision of other properties of security.

Mishra [21] indicated that in the present world of technology, security protocols built on strong cryptographic algorithms which attempt to defeat analysis patterns are common. With the major challenges of privacy, trustworthiness and security, a security protocol using minimal processor capacity and facilitating the targeted benefits of security is proposed. The chosen protocol works against the various security issues with the existing IoT protocols and is especially strong against severe attacks.

Wang et al. [29] proposed a self-trustworthy and secure Internet protocol (T-IP) for encrypted and authenticated network layer communications with the following advantages: (1) the IP address is self-trustworthy; (2) it has low connection latency and transmission overhead; (3) IP is stateless; and (4) it is compatible with the existing TCP/IP architecture.

Malina et al. [20] focused on the cryptographic mechanisms that could be useful and efficient on devices. They noted that the security solutions designed for IoT environments need to deal with heterogeneous entities with different specifications of software. Such devices use the constrained application protocol (CoAP) which provides authorization, authentication, confidentiality, data authenticity, freshness and integrity. By looking at the performance analysis of cryptographic primitives and memory limitations, they examine and discuss the applicability of privacy enhancing schemes and protocols.

Nguyen et al. [23] discussed the applicability and limitations of using IP-based Internet security protocols and other security protocols used in WSNs which have the potential to be used in IoT. Granjal et al. [12] also conducted a survey of the existing protocols and mechanisms to secure IoT communications. Their analysis indicates how the approaches in place ensure the security and protection of communications on the IoT. In Raza et al. [25], the focus is on exploring the option of IPsec use as a security mechanism for IoT. They present a 6LoWPAN/IPsec extension and indicate the approach's viability, finding that the IPsec is feasible for securing IoT.

Chen et al. [11] proposed a routing protocol based on node convergence degree and BCDTV. This protocol is used for cluster head election. The trust value is computed accounting the data transmitted, the forwarding rate of nodes and the amount of energy carried by the node. Based on these metrics, a five-stage cluster head evaluation scheme is created. The first stage gathers enerty of the cluster head nodes. Stage two, is about the communication metrics and the data metrices. In stage 3 all data are integrated to provide the convergence degree. Then, cluster head is elected and clusters are established. Finally the system will continue into transmission stage.

Arridha et al. [8] utilises data produced in IoT environment. They addressed the application of data analytics in IoT. The water condition monitoring system SEMAR is improved for analytics. This system explored the field of 4G transmission in marine areas. Most importantly, a real-time classifier based on decision tree algorithms.

Javanmardi et al. [14] computed a fuzzy reputation-based model to manage trust in P2P networks. Fuzzy logic is used based on values from node reputation to compute trust levels. The trust levels can be low, medium, or high to represent different set of ranking. The network are set in a clustered manner where the super peer engages with the nodes in its cluster and other super peers in other clusters.

Alsumayt et al. [6] explored the nature of IoT in MANETs. MANET nodes uses Internet Protocol for communication with each others. They proposed a trust value architecture, the Distributed cooperative trust-based intrusion detection (DICOTIDS) architecture. This architecture addresses different types of attack such as Dos attack and greyhole attack.

From the above and based on the analysis in Table 1, we conclude that although some trust or reputation systems have been proposed for IoT, the existing literature does not focus on countering key attacks in IoT trust systems, such as bad service providers, on-off attacks and con-behaviour attacks.

In the proposed solution, the bad service providers are defined as nodes providing service respond messages with incorrect structure, bad data or false data. The definition of a bad service provider is dependent on user perception. Nodes performing on-off attacks are bad service providers determined by time. For example, it could provide good service in the current time period but performing a bad service in the next time period. This change of behaviour made it harder to detect comparing to bad service providers. Similarly, a con-behaviour node provides good services to a group of nodes, but bad services to another group of nodes. In our system, this behaviour is realised with nodes providing good services within its cluster but providing bad services to node within another cluster.

Table 1 Comparative analysis of the existing literature from the perspective of countering attacks for IoT trust systems

References	Solution to counter on-off attacks in IoT trust systems	Solution to counter bad service providers in IoT trust systems	Approach to counter con-behaviour attack in IoT trust systems
Mosenia and Jha [22]	No	No	No
Alshehri et al. [5]	No	Yes	No
Alshehri and Hussain [3]	No	No	No
Lee et al. [16]	No	Yes	No
Ray et al. [24]	No	Yes	No
Kotis et al. [15]	No	Yes	No
Al-Fuqaha et al. [2]	No	No	No
Ahmed et al. [1]	No	No	No
Renubala and Dhanalakshmi [26]	No	No	No
Chen et al. [10]	No	No	No
Lize et al. [18]	No	Yes	No
Sirisala and Bindu [27]	No	Yes	No
Tuna et al. [28]	No	No	No
Lin et al. [17]	No	Yes	No
Ammar et al. [7]	No	No	No
Mishra [21]	No	No	No
Nguyen et al. [23]	No	No	No
Granjal et al. [12]	No	No	No
Raza et al. [25]	No	No	No
Chen et al. [11]	No	No	No
Javanmardi et al. [14]	No	No	No
Alsumayt et al. [6]	No	No	No
Proposed method	Yes	Yes	Yes

From the above evaluation of the existing literature, we conclude that in the existing literature there is no work to intelligently detect and counter bad service providers, "on-off" attacks and con-behaviour attack in IoT reputation systems. In order to address this issue, we take a systematic approach as follows:

(a) In Sect. 4, we propose a comprehensive fuzzy-logic based trust management approach based on clustering of IoT nodes to counter three different types of attacks in IoT trust systems, namely "on-off" attacks, con-behaviour attacks and bad service providers. We evaluate the performance of our proposed approach and the results are discussed in Sect. 6.
(b) To support the fuzzy-logic based trust management approach, we propose a HEXA decimal-based messaging system (based on TCP/IP) that can used to detect tampered messages in transit. This is presented in Sect. 3.
(c) In Sect. 5, we present the overview of the clustering-based trust protocol which uses the fuzzy-logic based trust management approach presented in Sect. 4.

In the next section we present a HEXA decimal-based messaging system for detecting tampering of messages between IoT nodes.

3 A secure HEXA decimal-based messaging system for tamper detection

The structure of a message is shown in Fig. 1 and Table 2. Each unit/code of the message is a two-digit hexadecimal number or an unsigned char with values from 0 to 255. *Data Length* refers to the length of the data section. *Check Code* is generated by processing other hexadecimals in the message. In our simulation, the *Check Code* is generated by adding the *Data* Sections together. If the result is greater than 255, reduce 256 from the result to avoid exceeding the numerical maximum of a two-digit hexadecimal 255 until it is smaller or equal to 255.

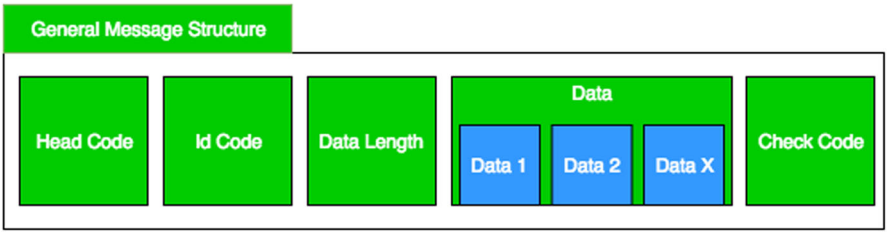

Fig. 1 General structure of a message

Table 2 A description of the general structure of a message

Code	Description	Length (bytes)
Head code	The head code is the same for all messages within the system	1
Id code	Determines which operation should be performed with the current message	1
Data length	Determines the length of the data section	1
Data	Data required for the operation performed by the current message	Depends on the operation
Check code	Calculated by the code before sending the message. Will be calculated again at the receiver node and compared to verify the validity of the message	1

The message system provides two extra layers of security. If the *head code* of a received message is wrong, the current message will be discarded, protecting the system from an outside source or malicious nodes. A different *check code* implies a wrong *check code* generation mechanism, which means the message is from an unsecure origin.

4 Fuzzy logic-based approach for countering attacks on IoT

This solution (Fuzzy-IoT) consists of five algorithms. Algorithm one is used to classify the trust score values into fuzzy sets. After determining the fuzzy sets, algorithm two use these fuzzy sets and classifies the cluster nodes into three categories: trusted, semi-trusted, non-trusted. These categories can restrict node interaction. Algorithm three uses a direct trust score, indirect trust score and routing score to calculate the trust value. Algorithm four uses this trust value to determine if a cluster node is able to change to another cluster. Finally, Algorithm five checks the current condition of the cluster nodes (CNs) and uses all the previous algorithms to update a new fuzzy status and new trust value. In conclusion, the fuzzy state will be used to limit node functionality and the calculated trust values will be used for clustering with trust boundaries stored on a certain master nodes.

Algorithm one uses fuzzy boundaries to determine a low, medium or high value for the three different trust scores. The whole system includes trust values from 0 to 1. In our case, fuzzy boundaries are defined as in Fig. 2.

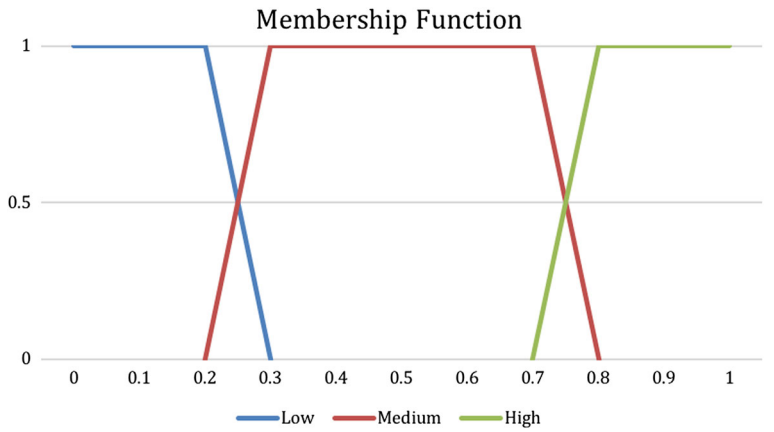

Fig. 2 Membership function of fuzzy logic

Algorithm one evaluates an input value with all three of these fuzzy sets, where obtaining a set of three 1 or 0 results represents membership or non-membership of the sets. Each of the values is tested with a set of three equations with an upper bound and lower bound. For example, with the medium number set, a value is compared that is between the lower bound of medium set Ml and the upper bound of medium set Mu.

$$Ml < value < Mu \tag{1}$$

Algorithm 1 Fuzzification Algorithm

Require: float Sc; upper boundary for low, Lu; upper and lower boundaries for medium and high, Mu, Ml, Hu, Hl
String $result \leftarrow$" "
if Sc <Lu **then**
$result \leftarrow$"1"
else
$result \leftarrow$"0"
end if
if Sc >Ml and Sc <Mu **then**
$result \leftarrow result+$"1"
else
$result \leftarrow result+$"0"
end if
if Sc >Hl and Sc <Hu **then**
$result \leftarrow result+$"1"
else
$result \leftarrow result+$"0"
end if
Return result

In our system, there will be three trust scores as components of the trust value. A direct score will be generated from the quality of a service response. When a new direct score is stored, the last one will become the new history score. Routing scores are generated by evaluating service responses from a node in a different cluster.

Placing these three scores into algorithm one generates the fuzzy membership for the three scores. Algorithm two uses the results of algorithm one to provide the fuzzy

state of a node. There are three fuzzy states. All nodes start with the non-restricted state "trusted". As more nodes give more trust scores, it could move down to more restricted states. A "semi-trusted" node can only provide services to nodes within the same cluster. "Semi-trusted" nodes are restricted to provide services to an outside node. A service request to a "semi-trusted" or "non-trusted" node will be blocked by its master node.

Algorithm two is based on logic. If the routing score (*Rr*) is low AND the direct score (*Rd*) is NOT low AND the past score (*Rp*) is not low, this node is *semi-trusted*. If *Rd* is low OR *Rp* is low, then this node is *non-trusted*. All nodes start in a *trusted state*. These states can only downgrade towards the *non-trusted* state.

Algorithm 2 Fuzzy Trust state detection

Require: fuzzification result of Direct, Past, Routing Scores, Rd, Rp, Rr;
if Rr is Low AND Rd is not Low AND Rp is not Low **then**
Return Semi-Trusted
end if
if Rd is Low OR Rp is Low **then**
Return Non-trusted
end if
Return Trusted

By directly using the average of the trust scores obtained from the other cluster nodes and other master nodes, a trust value can be obtained. Three direct scores, a past score and routing score coefficients *Cd*, *Cp* and *Cr* are values with a sum of 1. These coefficients are used to provide a weighted average of direct, past and routing scores (*Srd*, *Srp*, *Srr*). In our approach, we use an average as in Eq. 2 below, however a weighted approach could be used as well.

$$result = Cd \times Srd + Cp \times Srp + Cr \times Srr \tag{2}$$

Algorithm 3 Calculation of trust value

Require: Response, History, Routing Coefficients, Cd, Cp, Cr; Direct, Past, Routing Scores. Srd, Srp, Srr;
$result \leftarrow Cd \times Srd + Cp \times Srp + Cr \times Srr$
Return result

When a node's trust value is not within the boundaries of the current master node, Algorithm four uses the trust values calculated from algorithm three and checks if it is within the range of another master nodes' trust value boundaries (*Tnl* and *Tnu*). These boundaries can be requested from a super node (SN). The working of the algorithm for switching a node from one cluster to another by the master node (MN) of the cluster to which the node belongs is presented in Algorithm four below.

Algorithm 4 Request to switch cluster

Require: cluster node, CN; CN's trust value, Tcn; CN's Non-trusted status, NTs; Tcn; trust boundaries of new master node, Tnu, Tnl;
if $Tnl \leq Tcn \leq Tnu$ **then**
Return Request Granted
end if
Return Request Declined

Algorithm five specifies the process by which a master node checks the status of the cluster nodes and decides whether it needs to move them. The direct, past and routing scores (*Sdi*, *Spa*) are obtained from other cluster nodes and *Ssdi* and *Sspa* are summed.

$$Ssdi = \sum Sdi \tag{3}$$

$$Sspa = \sum Spa \tag{4}$$

The average of these scores, *Asdi* and *Aspa*, will then be calculated. Then, routing scores *Sro* are obtained from the other master nodes to calculate *Ssro*.

$$Ssro = \sum Sro \tag{5}$$

Again, an average *Asro* will be calculated. Algorithm two and three will be used to produce a fuzzy state (*NTs*) and a trust value (*Trust*), respectively. The fuzzy state will be broadcasted to other same-cluster nodes. The trust value obtained will be checked against the upper and lower trust boundaries (*Tu*, *Tl*) of the current master node. If it is not within the boundary, algorithm four will be run to check if this node can move to another cluster. The following flow chart (Fig. 3) demonstrates the process of Algorithm five.

Algorithm 5 Check current cluster node status

Require: List of Cluster nodes CNs; upper and lower trust boundaries of current master node, Tu, Tl; other master nodes, MNs
for all CN in CNs **do**
  for all other cluster nodes in CNs **do**
    get Direct and Past scores, Sdi, Spa
    $Ssdi \leftarrow Ssdi + Sdi$
    $Sspa \leftarrow Sspa + Spa$
  end for
  $Asdi \leftarrow Average(Ssdi)$
  $Aspa \leftarrow Average(Sspa)$
  for all other master nodes in MNs **do**
    Request Routing Scores, Sro;
    $Ssro \leftarrow Ssro + Sro$
  end for
  $Asro \leftarrow Average(Ssro)$
  $NTs \leftarrow Algorithm2(Asdi, Aspa, Asro)$
  broadcast NTs to same cluster neighbours
  $Trust \leftarrow Algorithm1(Cd, Cp, Cr, Asdi, Aspa, Asro)$
  if $Tl \leq Trust \leq Tu$ **then**
    Continue to next CN
  else
    for all MN in MNs **do**
      if Algorithm4(CN, Trust, NTs, trust boundaries of MN) is Granted **then**
        Continue to next CN
      end if
    end for
  end if
end for

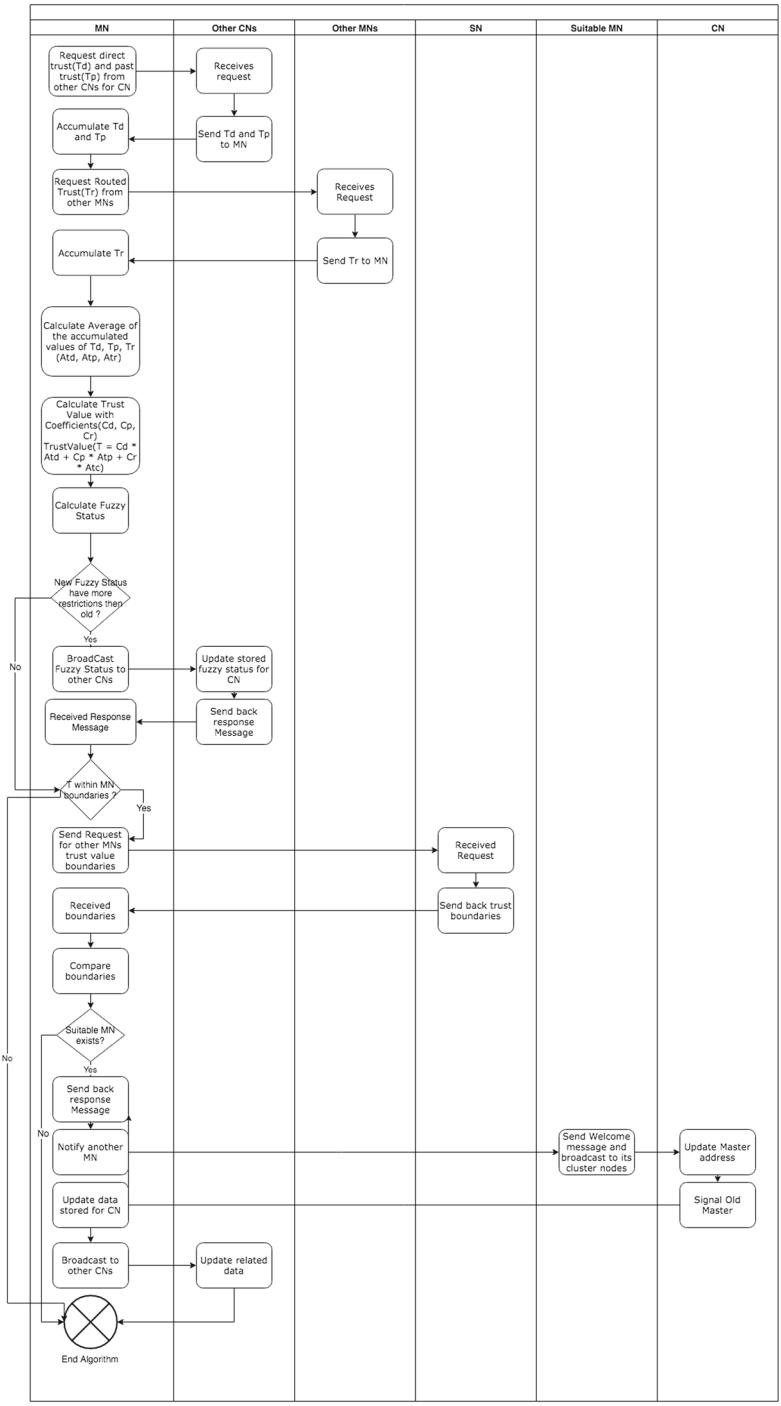

Fig. 3 Flowchart showing pictorially the working of the algorithm five

5 An intelligent security protocol

Figure 4 shows part of our security protocol. The *blue lines* are the connections between master nodes and its cluster nodes. The cluster nodes in the *blue cubes* can only make contact with the master node (*red cubes*) of its cluster. The super node can only be contacted by the master nodes. The *black lines* demonstrate basic same-cluster interactions. Cluster nodes can be classified into three categories with restrictions as shown in Table 3.

All cluster nodes have a fuzzy bank which stores the fuzzy status and node type of its neighbours. Initially, the fuzzy status of every neighbour of a node is trusted. For the same cluster request, the sender will firstly check the fuzzy status of the target. If the target is non-trusted, such as cluster node 3, the node will block itself from sending the request.

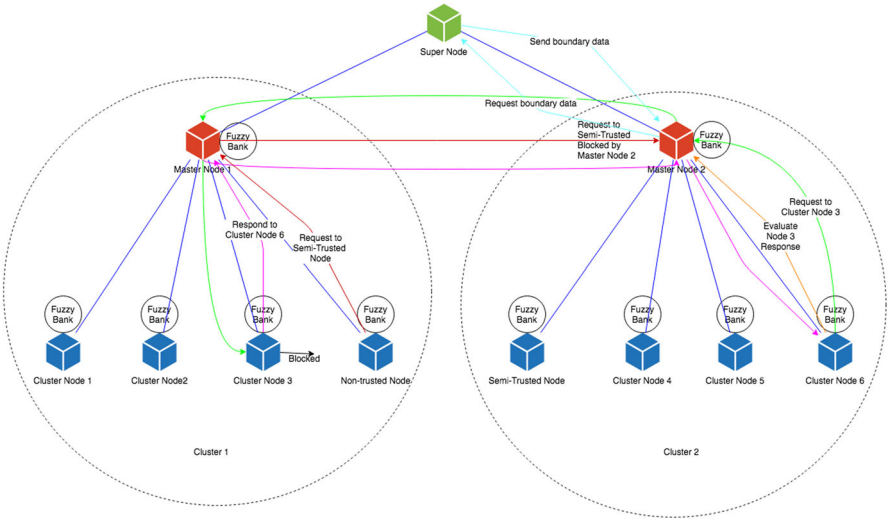

Fig. 4 Security protocols and message system mechanisms

Table 3 Types of nodes in the security protocol

Node types	Restrictions	Service quality
Normal nodes (trusted nodes)	No restrictions	Provides an acceptable quality service
Semi-trusted nodes	Cannot send a routed service response to a service request from a node in another cluster. Able to send out service requests.	Reliable service for the same cluster nodes. Bad quality service to nodes in another cluster.
Non-trusted nodes	Cannot respond to any service requests. Able to send out service requests	Bad quality service

A service request from a node to a node in another cluster will be routed through the master nodes. The *green line* shows a request reaching from cluster node 6 to cluster node 3, routed from master node 2 to master node 1 and finally to cluster node 3. The *purple lines* show a response which is sent back from cluster 3 to cluster 6 routed by master node 1 and master node 2. Finally, the *orange line* shows a score from cluster node 6 being sent to master node 2 for cluster node 3's services.

The fuzzy bank of master nodes stores the fuzzy status and trust values of its cluster nodes. It also stores the scores rated by the cluster nodes for nodes in another cluster. The *red lines* show the non-trusted nodes sending a service request to the semi-trusted nodes. Master node 2 checks the fuzzy status of the semi-trusted node and blocks the request.

The *light blue lines* show that the super node provides trust value boundaries for the master nodes to be used in a future cluster change algorithm (Algorithm 5).

6 Simulation, node mechanisms, results and analysis

This section presents the details of simulation, node mechanisms and the analysis of results.

6.1 Simulation settings: basic concept

The simulation is run with the Cooja simulator from the Contiki system. Contiki is an event-driven and lightweight system for the IoT. In this simulation, Rime addresses from Contiki are used for node identification. By default, Rime addresses can be 2 bytes or 8 bytes. Considering the small size of our clusters, 2-byte addresses will be sufficient. There are two major sets of simulations: the base case with the fuzzy detection mentioned in the node mechanism section; and the second case without fuzzy detection. These two sets of simulations are further developed in three different simulation scenarios. The first scenario only has 15 nodes to test the basic counter-attack concept of the algorithm. The second scenario utilises 200 nodes to ensure the algorithm is functional under a large number of nodes. The final scenario contains 2000 nodes showing the algorithm's performance on a large-sized network.

In our algorithm, the function of a super node is to send trust boundaries to the master nodes. Because this is a small system, we simplified the simulation and stored the boundaries on the master nodes. Three malicious nodes are in the system for detection: (1) a bad service provider which sends low scoring service responses; (2) an on-off attack node which provides a bad service in a certain cycle of time; and (3) a con-behaviour node which sends a bad service response redirected to nodes within other clusters.

Table 4 Basic concept node initial clustering

Master node	Cluster nodes
1	4,5,6,7
2	8,9,10,11
3	12,13,14,15

Table 5 Basic concept node properties

Node	Rime address	Node type
1	1.0	Master node
2	2.0	Master node
3	3.0	Master node
4	4.0	Cluster node
5	5.0	Cluster node
6	6.0	Cluster node
7	7.0	Bad service provider
8	8.0	Cluster node
9	9.0	Cluster node
10	10.0	Cluster node
11	11.0	On-off attack node
12	12.0	Cluster node
13	13.0	Cluster node
14	14.0	Cluster node
15	15.0	Con-behaviour node

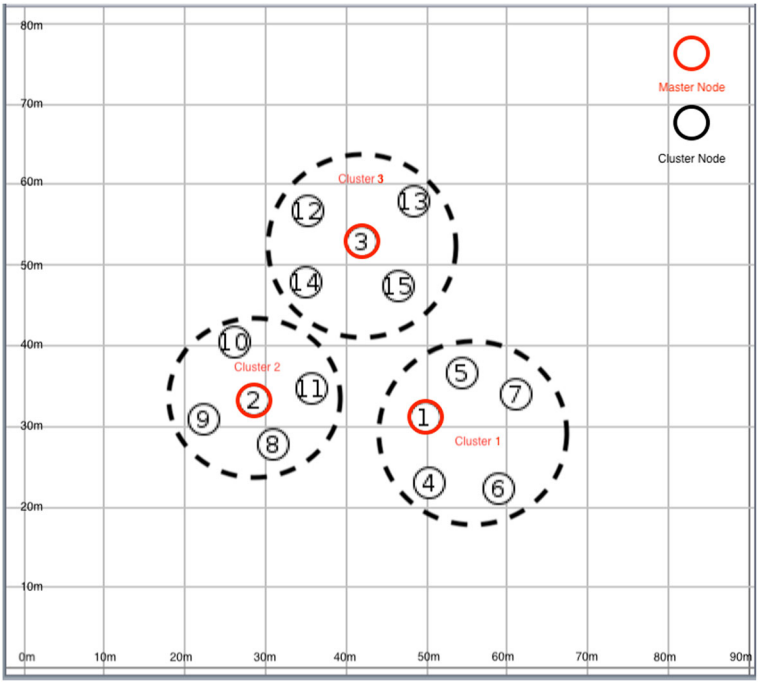

Fig. 5 Basic concept initial simulation setting

Table 4 demonstrates the initial clustering of the 15 nodes. Node 1, Node 2 and Node 3 act as master nodes for the three initial clusters. Table 5 shows the details of the types of nodes within the simulation. Nodes 7, 11 and 15 are malicious nodes

Table 6 Basic concept base case parameters

Parameter	Value
Number of nodes	15
Number of clusters	3
Simulation time	60 s
Simulation area	90 m × 80 m
Fuzzy trigger	On

Table 7 Basic concept non-fuzzy case parameters

Parameter	Value
Number of nodes	15
Number of clusters	3
Simulation time	60 s
Simulation area	90 m × 80 m
Fuzzy trigger	Off

acting as a bad service provider providing bad responses, a node performing on-off attacks and a node performing con-behaviour attacks, respectively. These nodes are distributed accross a 90 m × 80 m surface into three clusters as shown in Fig. 5.

This simulation consists of two cases. Table 6 details the base case. The base case takes place on a 90 m × 80 m surface for 60 s with the *Fuzzy Trigger* on. If the *Fuzzy Trigger* is on, the cluster nodes and master nodes block requests sent to *Non-trusted* and *Semi-trusted* nodes as mentioned in Sect. 5. Table 7 shows settings for the second case. The only difference between the cases is the second case runs with the *Fuzzy Trigger* off. In this case, the master nodes and cluster nodes will not block any requests considering the fuzzy status.

6.2 Simulation settings: 200 nodes

The second simulation scenario is used to test the proposed system in a network of 200 IoT nodes with 60 being malicious.

Table 8 shows the initial cluster settings of the network. The first 20 nodes in the network are all master nodes, each carrying nine cluster nodes in their cluster. Table 9 shows the node property composition of every cluster. In every cluster, the last three nodes are malicious nodes. These three nodes are the bad service povider, on-off attack node and con-behaviour node. A topology of this case is shown in Fig. 6.

This simulation scenario also consists of two cases. Table 10 demonstrates the base case. The base case takes place on a 100 m × 100 m surface for 60 s with the *Fuzzy Trigger* on. Table 11 shows settings for the second case with the *Fuzzy Trigger* off.

Table 8 200 Nodes IoT network node initial clustering

Master node	Cluster nodes
1	21–29
2	30–38
3	39–47
4	48–56
5	57–65
6	66–74
7	75–83
8	84–92
9	93–101
10	102–110
11	111–119
12	120–128
13	129–137
14	138–146
15	147–155
16	156–164
17	165–173
18	174–182
19	183–191
20	192–200

Table 9 200 Node IoT network node properties

Node	Rime address	Node type
1–20	1.0–20.0	Master node
The first six cluster nodes in each cluster		Cluster node
The seventh cluster node in each cluster		Bad service provider
The eighth cluster node in each cluster		On-off attack node
The ninth cluster node in each cluster		Con-behaviour node

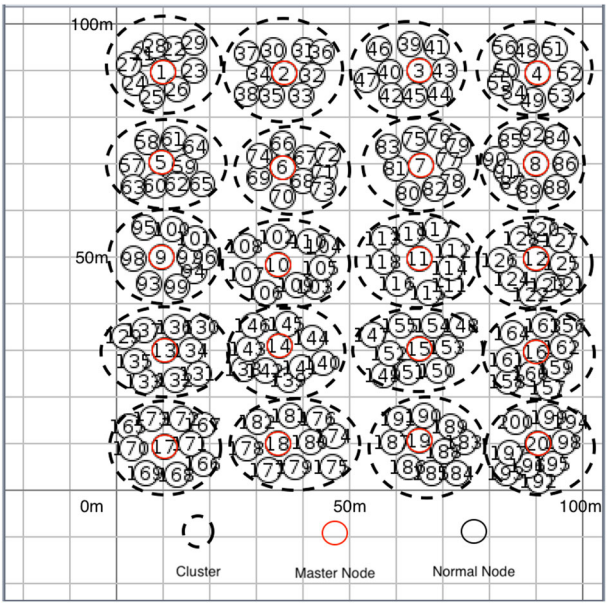

Fig. 6 200 Node IoT network initial simulation setting

Table 10 200 Node IoT network base case parameters

Parameter	Value
Number of nodes	200
Number of clusters	20
Simulation time	60 s
Simulation area	100 m × 100 m
Fuzzy trigger	On

Table 11 200 Node IoT network non-fuzzy case parameters

Parameter	Value
Number of nodes	200
Number of clusters	20
Simulation time	60 s
Simulation area	100 m × 100 m
Fuzzy trigger	Off

6.3 Simulation settings: large IoT network

The third simulation scenario is used to test the proposed system in a large scale IoT network with 2000 nodes to prove the proposed approach is able to scale to any number of IoT nodes. Of these 2000 nodes, 60 are malicious.

Table 12 Large IoT network node initial clustering

Master node	Normal node number	Cluster property	Description
1–20	6	Malicious cluster	Last three nodes are bad service, on-off, and Con-behaviour node
21–200	9	Normal cluster	All nodes are not malicious

Fig. 7 Large IoT network sample malicious cluster setting

Table 12 presents the initial property of the clusters. The first 20 clusters are malicious clusters, which each contains three malicious nodes. These clusters have a similar topology as Fig. 7. The 21th to the final cluster are normal clusters with all non-malicious cluster nodes. Figure 8 demonstrates this setting. Figure 9 reveals a big picture of cluster distribution for the large IoT network with 2000 node. Each cluster contains a master node and nine cluster nodes.

Table 13 demonstrates the base case of this simulation scenario. The base case takes place on a 300 m × 150 m surface for 60 s with the *Fuzzy Trigger* on. Table 14 shows settings for the second case with the *Fuzzy Trigger* off.

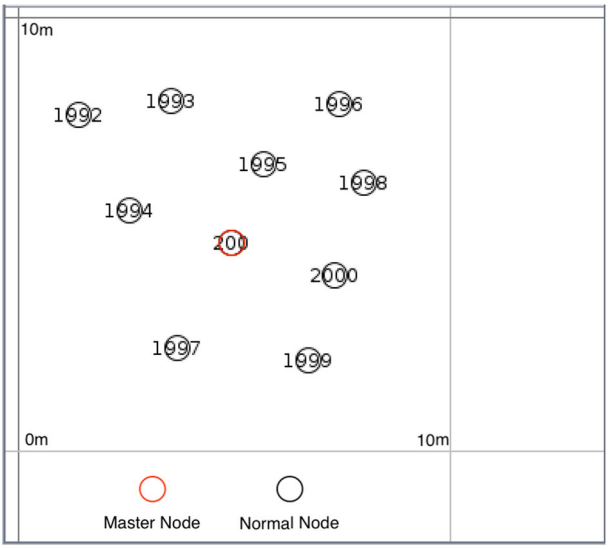

Fig. 8 Large IoT network sample normal cluster setting

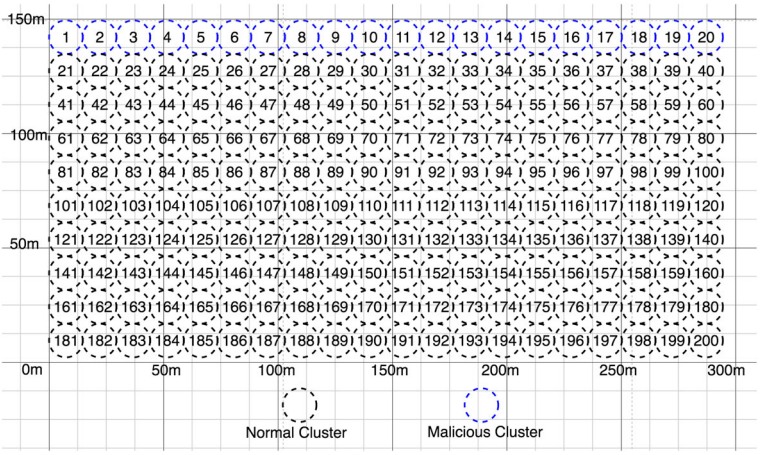

Fig. 9 Large IoT network initial simulation setting

6.4 Node mechanisms

A node can transmit a message to a node within its cluster, but it needs a master node to redirect a message to another master node, then to a node within another cluster. For direct transmission, if the fuzzy status of the target node stored on the sending node is non-trusted, the sending node will stop sending the message. For a redirected message, if the master node of the target determines the cluster node is semi-trusted or non-trusted, it will block the message immediately.

Table 13 Large IoT network base case parameters

Parameter	Value
Number of nodes	2000
Number of clusters	200
Simulation time	60 s
Simulation area	300 m × 150 m
Fuzzy trigger	On

Table 14 Large IoT network non-fuzzy case parameters

Parameter	Value
Number of nodes	2000
Number of clusters	200
Simulation time	60 s
Simulation area	300 m × 150 m
Fuzzy trigger	Off

6.5 Results and analysis

An analysis of the experimentation resulted in the following: Three bar graphs to measure the time is takes to intelligently detect the nodes carrying out three types of attack - bad service provider, on-off attack and con-behaviour attack:

(i) A performance evaluation and comparison to detect on-off attacks. The performance evaluation and comparison is carried out for both fuzzy and non-fuzzy cases.
(ii) A performance evaluation and comparison to detect con-behaviour at- tacks. The performance evaluation and comparison is carried out for both fuzzy and non-fuzzy cases.
(iii) Three diagrams comparing the average trust values for fuzzy and non- fuzzy cases. The first has similar settings to prove the basic concept. The second diagram is obtained in an environment with 200 nodes. The final diagram is obtained in an environment with 2000 nodes. The second and third diagrams are used to prove that the proposed algorithms can operate with a large number of IoT nodes.

The bar graph in Fig. 10 shows the number of rounds required to detect a certain malicious node carrying out malicious attacks. We define a "Round" as a master node finishing one check of the current status of all its cluster nodes (algorithm 5). This process includes gathering the trust scores from the master nodes and cluster nodes, calculating the trust values and fuzzy status, assigning a fuzzy status and finally moving out of the trust value boundary cluster nodes to a suitable master node. As can be seen in Fig. 10, all of the bad service providers, the on-off attack nodes and the con-behaviour attack nodes are detected during round 1 of the master nodes' cycle. This demonstrates that this algorithm is quite efficient in detecting these malicious node types.

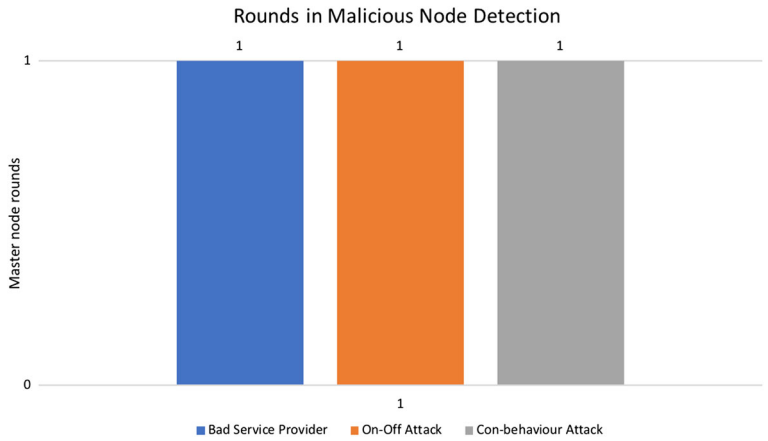

Fig. 10 Rounds in malicious node detection

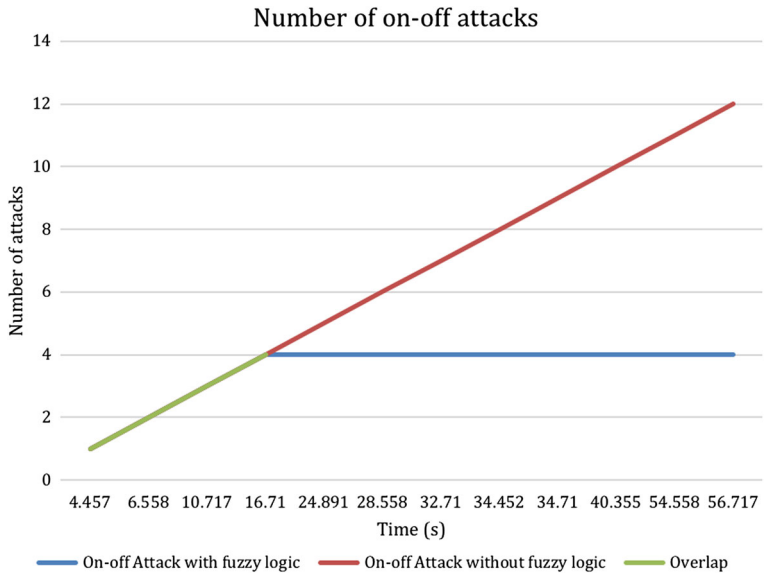

Fig. 11 Number of on-off attacks

Figure 11 compares the on-off attacks with and without fuzzy logic. Between 0 and 16.71 s, the number of on-off attacks continues to increase. However, using the fuzzy mechanisms proposed in this paper, no new on-off attacks occur after 16.71 s. In the case without fuzzy mechanisms, these attacks won't be detected, as indicated by the *orange line* on-off attacks which shows a constant increase in these attacks. This demonstrates that the fuzzy mechanism presented in this paper is effective in detecting on-off attacks and is able to block the attack after the initial bootstrapping time.

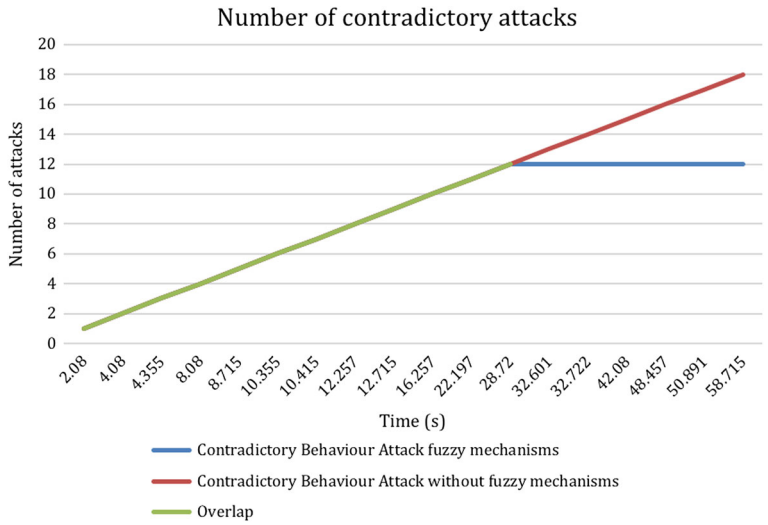

Fig. 12 Number of contradictory behaviour attacks

Similar to on-off attacks, in Fig. 12 a contradictory behaviour attack node is determined at 28.72 s using the approach proposed in this paper. The *blue line* indicates that the number of contradictory behaviour attacks with fuzzy mechanisms does not increase after 28.72 s as no new attacks are performed. The *red line* indicating contradictory behaviour attacks without fuzzy mechanisms continues to increase as the attacks are not blocked. This demonstrates the effectiveness of the proposed protocol to block contradictory behaviour attacks.

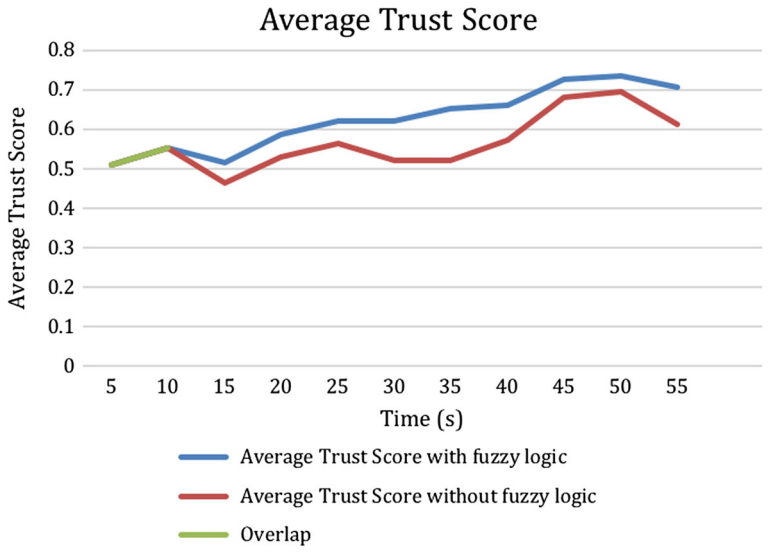

Fig. 13 Average trust score of all the IoT nodes during the simulation (n = 15 nodes)

Figure 13 shows the average ratings for the services. In our model, a node gives a score after receiving a service response which is the basis of the trust value. This score represents the service quality of the received service response. The lines for the fuzzy mechanisms and without the fuzzy mechanisms diverge when node 7, the bad service provider, is detected by master node 1. A larger divergence occurs around 15 s and 30 s when node 11, the on-off attack node, and node 15, the con-behaviour node, are detected, respectively. This shows there is an increase in the average service quality of the whole system when a fuzzy mechanism is added.

In this scenario of 200 nodes (Fig. 14) fuzzy logic still contributes to trust value. This figure is similar to the base case scenario above (Fig. 13). The only difference will be the initial trust value and the difference between fuzzy trust value, non-fuzzy trust values. The base scenario have three malicious node in twelve cluster nodes. The proportion of malicious node is 25%. While in the case of 200 nodes, there are 60 malicious nodes in 10 cluster nodes. This is 33.3% of malicious nodes. With a greater number of malicious nodes the initial trust value of 200 nodes is slightly lower than the base scenario. In the base scenario, one malicious node represents a third of all malicious nodes, on the other hand, one malicious node only represents 1.67% of total malicious nodes in the 200 nodes network. In this case, although there are more malicious nodes in every cluster of the 200 node network, the difference between the two lines is smaller than the base scenario. Figure 15 shows that the average trust of the large IoT network is similar. Only 3.33% of the 1800 cluster nodes are malicious. This equates to a significantly higher initial average trust value than the other two scenarios, but there is a smaller difference between the two cases of fuzzy logic, as each detection of a malicious nodes is not that significant.

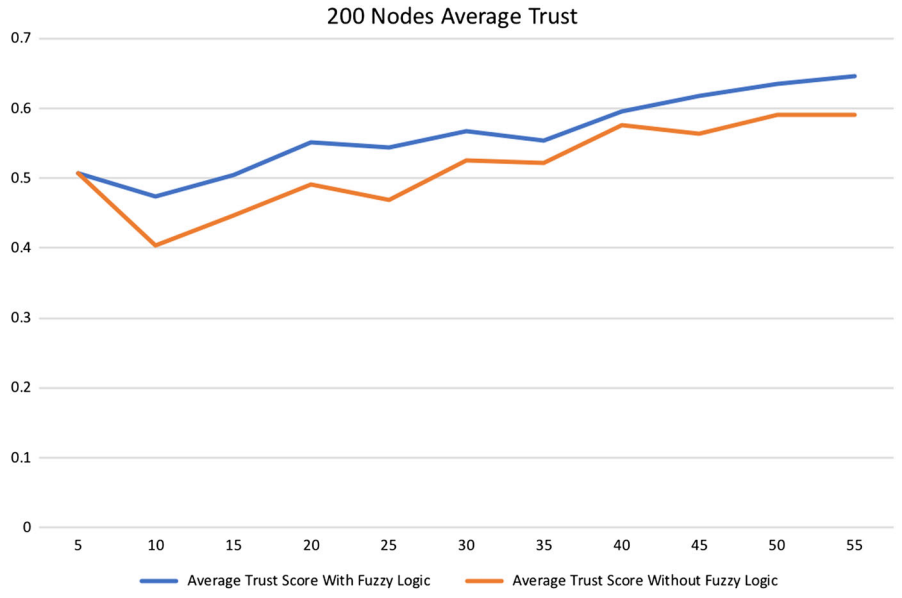

Fig. 14 Average trust score of all the IoT nodes during the simulation (n = 200 nodes)

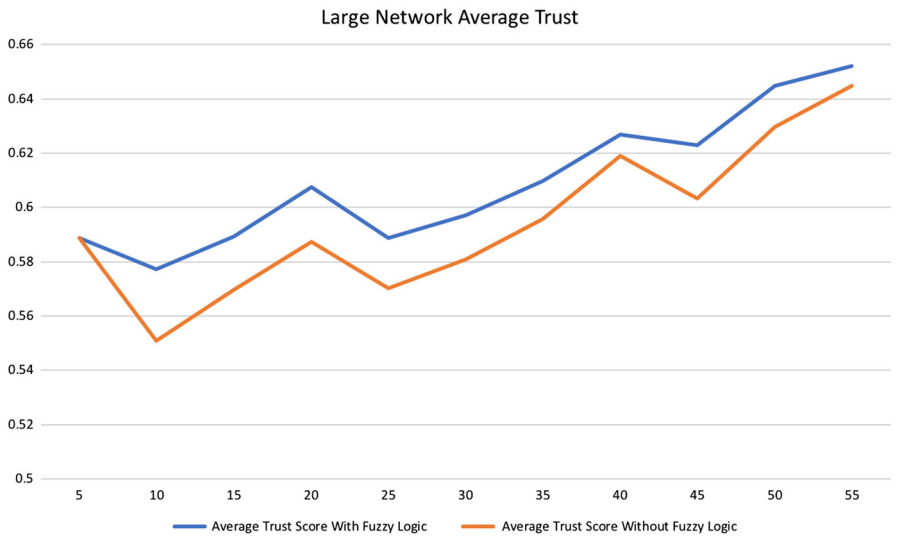

Fig. 15 Average trust score of all the IoT nodes during the simulation (n = 2000 nodes)

7 Conclusion

The security and trustworthiness of IoT nodes is a critical and pressing issue and has received a lot of attention in the existing literature. An IoT network can scale up or down dynamically with time and with new nodes continuing joining it and existing nodes exiting the network. A trust solution should be able to adapt to these changes in network size. To address this critical issue, in this paper, we propose a cluster-based fuzzy-logic approach, in which existing nodes are grouped into clusters. We proposed novel approaches to address three critical issues to enable reliable cluster-based trust management in IoT. First, we proposed a protocol using fuzzy logic that is able to detect on-off attacks, contradictory behaviour attacks and other malicious nodes. Second, we demonstrated how this approach handles IoT nodes to maintain part of its function using fuzzy logic. Third, we proposed a secure message system for IoT nodes using hexadecimal values with a structure similar to serial communication. We carried out extensive experimentation and evaluation of the results under varying network sizes to capture the scalability of our proposed approach and also to measure its efficiency in detecting malicious nodes, such as bad service providers, on-off attacks and con-behaviour attacks. We found from the experimentation results that our approach is very effective in identifying the malicious nodes in the network within a set simulation time frame of 60 s. Finally, we found that using our approach, the computed average trust value converges quickly to the actual average trust value of the network.

References

1. Ahmed A, Bakar KA, Channa MI, Haseeb K, Khan AW (2015) TERP: a trust and energy aware routing protocol for wireless sensor network. IEEE Sens J 15(12):6962–6972. https://doi.org/10.1109/JSEN.2015.2468576

2. Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M (2015) Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor 17(4):2347–2376. https://doi.org/10.1109/COMST.2015.2444095
3. Alshehri MD, Hussain FK (2017) A centralized trust management mechanism for the internet of things (CTM-IoT). In: International conference on broadband and wireless computing, communication and applications. Springer, Cham, pp 533–543. https://doi.org/10.1007/978-3-319-69811-3_48
4. Alshehri MD, Hussain FK (2015) A comparative analysis of scalable and context-aware trust management approaches for internet of things. In: International conference on neural information processing, 2015. Springer, pp 596–605. https://doi.org/10.1007/978-3-319-26561-2_70
5. Alshehri MD, Hussain FK, Hussain OK (2018) Clustering-driven intelligent trust management methodology for the internet of things (CITM-IoT). Mobile networks and applications, 1–13. https://doi.org/10.1007/s11036-018-1017-z
6. Alsumayt A, Haggerty J, Lotfi A (2017) Using trust to detect denial of service attacks in the internet of things over MANETs. Int J Space-Based Situated Comput 7(1):43–56. https://doi.org/10.1504/IJSSC.2017.10004987
7. Ammar M, Russello G, Crispo B (2018) Internet of things: a survey on the security of IoT frameworks. J Inf Secur Appl 38:8–27. https://doi.org/10.1016/j.jisa.2017.11.002
8. Arridha R, Sukaridhoto S, Pramadihanto D, Funabiki N (2017) Classification extension based on IoT-big data analytic for smart environment monitoring and analytic in real-time system. Int J Space-Based Situated Comput 7(2):82–93. https://doi.org/10.1504/IJSSC.2017.10008038
9. Chasaki D, Mansour C (2015) Security challenges in the internet of things. Int J Space-Based Situated Comput 5(3):141–149. https://doi.org/10.1504/IJSSC.2015.070945
10. Chen D, Chang G, Sun D, Li J, Jia J, Wang X (2011) TRM-IoT: a trust management model based on fuzzy reputation for internet of things. Comput Sci Inf Syst 8(4):1207–1228. https://doi.org/10.2298/CSIS110303056C
11. Chen L, Qi X, Liu L, Zheng G (2017) A security routing protocol based on convergence degree and trust. Int J Grid Utility Comput 8(1):38–45. https://doi.org/10.1504/IJGUC.2017.10003008
12. Granjal J, Monteiro E, Silva JS (2015) Security for the internet of things: a survey of existing protocols and open research issues. IEEE Commun Surv Tutor 17(3):1294–1312. https://doi.org/10.1109/COMST.2015.2388550
13. Hossain MM, Fotouhi M, Hasan R (2015) Towards an analysis of security issues, challenges, and open problems in the internet of things. In: Services (SERVICES), 2015 IEEE World Congress on, 2015. IEEE, pp 21–28. https://doi.org/10.1109/SERVICES.2015.12
14. Javanmardi S, Shojafar M, Shariatmadari S, Ahrabi SS (2014) Fr trust: a fuzzy reputation-based model for trust management in semantic p2p grids. Int J Grid Util Comput 6(1):57–66. https://doi.org/10.1504/IJGUC.2015.066397
15. Kotis K, Athanasakis I, Vouros GA (2018) Semantically enabling IoT trust to ensure and secure deployment of IoT entities. Int J Internet Things Cyber-Assur 1(1):3–21. https://doi.org/10.1504/IJITCA.2018.090158
16. Lee J-Y, Lin W-C, Huang Y-H (2014) A lightweight authentication protocol for internet of things. In: Next-generation electronics (ISNE), 2014 international symposium on, 2014. IEEE, pp 1–2. https://doi.org/10.1109/ISNE.2014.6839375
17. Lin J, Yu W, Zhang N, Yang X, Zhang H, Zhao W (2017) A survey on internet of things: architecture, enabling technologies, security and privacy, and applications. IEEE Internet Things J 4(5):1125–1142. https://doi.org/10.1109/JIOT.2017.2683200
18. Lize G, Jingpei W, Bin S (2014) Trust management mechanism for internet of things. China Commun 11(2):148–156. https://doi.org/10.1109/CC.2014.6821746
19. Mahalle PN, Thakre PA, Prasad NR, Prasad R (2013) A fuzzy approach to trustbased access control in internet of things. In: Wireless communications, vehicular technology, information theory and aerospace and electronic systems (VITAE), 2013 3rd international conference on, IEEE, pp 1–5. https://doi.org/10.1109/VITAE.2013.6617083
20. Malina L, Hajny J, Fujdiak R, Hosek J (2016) On perspective of security and privacy-preserving solutions in the internet of things. Comput Netw 102:83–95. https://doi.org/10.1016/j.comnet.2016.03.011
21. Mishra S (2015) Network security protocol for constrained resource devices in internet of things. In: India conference (INDICON), 2015 annual IEEE, 2015. IEEE, pp 1–6. https://doi.org/10.1109/INDICON.2015.7443737

22. Mosenia A, Jha NK (2017) A comprehensive study of security of internet-of-things. IEEE Trans Emerg Top Comput 5(4):586–602. https://doi.org/10.1109/TETC.2016.2606384
23. Nguyen KT, Laurent M, Oualha N (2015) Survey on secure communication protocols for the internet of things. Ad Hoc Netw 32:17–31. https://doi.org/10.1016/j.adhoc.2015.01.006
24. Ray BR, Abawajy J, Chowdhury M (2014) Scalable RFID security framework and protocol supporting internet of things. Comput Netw 67:89–103. https://doi.org/10.1016/j.comnet.2014.03.023
25. Raza S, Duquennoy S, Höglund J, Roedig U, Voigt T (2014) Secure communication for the internet of thing: a comparison of link-layer security and IPsec for 6LoWPAN. Secur Commun Netw 7(12):2654–2668. https://doi.org/10.1002/sec.406
26. Renubala S, Dhanalakshmi K (2014) Trust based secure routing protocol using fuzzy logic in wireless sensor networks. In: Computational intelligence and computing research (ICCIC), 2014 IEEE international conference on, 2014. IEEE, pp 1–5. https://doi.org/10.1109/ICCIC.2014.7238435
27. Sirisala N, Bindu CS (2015) Uncertain rule based fuzzy logic QoS trust model in manets. In: Advanced computing and communications (ADCOM), 2015 international conference on, Chennai. IEEE, pp 55–60. https://doi.org/10.1109/ADCOM.2015.17
28. Tuna G, Kogias DG, Gungor VC, Gezer C, Taşkın E, Ayday E (2017) A survey on information security threats and solutions for machine to machine (M2M) communications. J Parallel Distrib Comput 109:142–154. https://doi.org/10.1016/j.jpdc.2017.05.021
29. Wang X, Zhou H, Su J, Wang B, Xing Q, Li P (2018) T-IP: a self-trustworthy and secure internet protocol. China Commun 15(2):1–14. https://doi.org/10.1109/CC.2018.8300267
30. Yao X, Wang L (2017) Design and implementation of IOT gateway based on embedded μTenux operating system. Int J Grid Util Comput 8(1):22–28. https://doi.org/10.1504/IJGUC.2017.10003004