An Efficient and Secure Fog Based Routing Mechanism in IoT Network

Tauqeer Safdar Malik 1,†, Jawad Tanveer 2,†, Shahid Anwar 3, Muhammad Rafiq Mufti 4,©, Humaira Afzal 3 and Ajung Kim 5,*

Abstract: The Internet of Things (IoT) networks are the most prone to internal as well as external attacks in addition to energy consumption problems. Conventional security solutions are not able to address these issues effectively due to the limited resources of sensor nodes participating in IoT communications. In this work, an Efficient and Secure Fog Based Routing Mechanism (ESFRM) is proposed to protect the network from faulty internal as well as external attacks. Every node participating in IoT communications calculates the comprehensive trust value of the next intermediate node which is the addition of direct trust, indirect trust and energy trust values before forwarding the data. This comprehensive trust value is then compared with the comprehensive threshold trust value to decide whether the particular node is a rogue node or a valid normal node. Further, an enhanced RSA (Rivest, Shamir, Adleman) algorithm is implemented to provide three levels of data security from Cluster Head (CH) to fog node, from fog node to cloud server and directly from CH to cloud server. For this purpose, an efficient CH selection technique is also proposed in this work. The proposed methodology is compared with the Secure Energy-efficient Fog-based Routing (SEFR) protocol and Trust-aware Secure Routing Protocol (TSRP). The evaluation results show that the proposed ESFRM outperforms the conventional schemes with respect to energy consumption, malicious node detection and transmission rate.

Keywords: Internet of Things (IoT); cloud-native fog computing; advanced algorithms; wireless communications; security in IoT

MSC: 62F99; 65C99

1. Introduction

The Internet of Things (IoT) network is made up of small devices commonly known as sensor nodes that are used in physical and environmental conditions to monitor a wide range of parameters including temperature, pressure, light, heat, motion, sound, vibration and many more. These sensor nodes are often deployed in risky and unattended territory where they accumulate myriad types of environmental data and transmit it to the base station or sink node via hop-by-hop or intermediate nodes for further processing [1–7]. The IoT networks have been the buzz of the contemporary period due to their amazing effects that have taken the globe by storm. In order to deal with diverse problems, the sensor nodes are now being widely employed, thereby jacking up the number of devices to manifold.
Contrary to the past, where human resources were needed to monitor environmental factors in stable areas, the unsteady areas or regions used to remain unrevealed due to potential dangers and the incapacity of people to access the extremely volatile locations. Due to this, the sensor nodes are now used in these situations to lessen the difficulties faced by the general public. However, sensor nodes are installed in dangerous/hostile and even unattended premises where they are more prone to various kinds of attacks \cite{8-10}. Moreover, owing to the absence of any predefined topology in the IoT network, every node requires the capacity of routing for data forwarding. As a result, the nodes are increasingly exposed to myriad routing assaults \cite{11}. Although a lot of traditional cryptographic, detecting malicious nodes and routing algorithms \cite{12-20} have been designed to make IoT networks secure and safe these algorithms demand heavy computations/calculations and processing which cause lots of energy utilization. In addition to this, the cryptography and authentication-based algorithms work only on the IoT network comprising all the trustworthy and cooperative nodes, therefore, they are not suitable to cope with internal node attacks launched by malicious and faulty sensor nodes \cite{21-24}. In order to detect and remove faulty/malicious nodes to prevent these rogue nodes from launching internal node attacks within the network, trust-based security mechanisms have been proven to be most efficient and effective \cite{25}. Aside from the internal node attacks in IoT networks, efficient energy utilization is also a serious factor to consider since it has a significant influence on network performance. In addition to energy constraints, the storage and processing capacities of sensor nodes are also limited in fog cloud environment. In order to address the issues faced in the cloud environment and to alleviate the strain on cloud servers, the scientific community has developed a new idea called fog extending the functionality of cloud computing. The fog processing is a sort of distributed processing paradigm in which some data processing, calculation, and storage are carried out between the devices that generate the data on the cloud server \cite{26}. These objects, often referred to as fog devices, are positioned close to the sensor nodes that produce data. Therefore, the data that needs an immediate reply is analyzed, computed and processed here at fog devices and finally, the result is sent back to the desired location. The data that is frequently needed is also retained at fog devices, and the remaining information is sent to the cloud server for further processing in addition to long-term storage. This capability significantly reduces the workload on the server, bandwidth use, data dropping ratio, and latency rate and serves as a crucial tool to minimize response rate \cite{27,28}.

Since the fundamental objective of sensor nodes in fog-based IoT is to communicate the sensed data towards the intended destination, therefore, secure and efficient routing is a dire requirement to make the entire network secure, safe and reliable from internal as well as external attacks. Moreover, the data is most susceptible to disruption by intruders and even this data can be subverted by malicious/rogue intermediate nodes when sensed facts, figures, or evidence are communicated by participating member nodes to Base Station (BS) or sink node and then subsequently to fog node or cloud server. Rogue/malicious intermediary nodes perform a number of assaults to block the data from getting to its final destination. Examples of possible internal threats to the IoT include black hole attacks, sinkhole attacks, selective forwarding, Sybil attacks, warm hole attacks, and HELLO flood assaults, among others.

The main contribution of this paper aiming at making the entire network safe and secure from malicious internal node attacks, which are based upon trust awareness, is discussed below to detect and expel faulty/malicious nodes from the IoT network.

- The comprehensive trust value of each IoT network is computed which is comprised of direct trust, indirect trust and energy trust value.
- In order to identify and remove malicious nodes as quickly as possible, the volatilization factor has been included in the selection of Cluster Head (CH) to minimize energy utilization and to manage the densely IoT network.
- Aside from this, an enhanced RSA algorithm has also been employed in the proposed paradigm to protect data originality from external attackers.
The remainder of this paper is structured as follows: The relevant work is described in Section 2. Section 3 explains the proposed model of trust-aware security, encryption algorithm and cluster head selection technique. Section 4 shows the simulations with results discussion and finally in Section 5 conclusion is given.

2. Literature Review

A hybrid offline and online encryption approach to implement digital signature and encryption is demonstrated to speed up packet delivery and decrease communications problems in [29]. The high computing complexity and uncertain forms of defendable assaults are mostly its disadvantages. A novel encryption technique utilizing elliptic curve cryptography (ECC) and homomorphic encryption is used to provide the safe transmission of data in the clustering of WSNs [30]. It is used to outline the kinds of assaults, including hello flood, denial-of-service and compromised cluster heads. Still, this approach results in a high incidence of packet loss and a significant delay. The hello flood and selective forwarding attacks are successfully thwarted by creating unique 176-bit keys for each sensor by using the ECC technique to produce binary strings for each sensor and the rate of packet loss and the latency are both decreased in [31]. However, after 1000 rounds of the method, each node’s residual energy fluctuates significantly, which results in uneven energy usage. A better approach is put out to expedite multiuser message broadcasting authentication based on the four components such as system activation, user addition, multiuser broadcast authentication and user revocation to enable safe data transport in RSA [32]. A trust-based drone energy-saving data-collecting strategy to employ the drone path’s quadratic optimization algorithm to identify routing paths is presented in [33]. Additionally, trust inference and evolve procedures are also used to determine the sensor node’s level of trust. As a result, it can efficiently determine an ideal data-gathering route and balance the network’s energy usage. In order to save energy, the Beta Reputation and Direct Trust (BRDT) approach uses the beta and direct trust paradigm for safe and reliable communication in IoT networks [34]. Nevertheless, there are frequently too many cluster heads as a result of the wide communication range overlaps, which causes a lot of energy wastage. Furthermore, BRDT does not specify the defendable assaults. To counter black-hole attacks, Grade Trust, a secured routing system based on the trust levels of nodes, is discussed in [35]. Where only a black-hole attack can be thwarted by Grade Trust, despite an increase in packet delivery ratio. In [36], a secure routing technique based on clustering is implemented an energy-efficient clustering technique to encrypt the data of the network using a trusted hardware module which can successfully fend off several assaults such as data confidentiality and integrity attacks as well as comparison node attacks. However, the cluster head nodes must have equipment for persistent energy supply which raises the bar for the IoT configuration. A trust-based energy-preserving multi-hop routing system is proposed to combine encryption and a trust management-based protocol [37]. There are some inaccuracies when computing the trust values of neighbor nodes because of indirect trust value calculation. The direct and indirect trust values of nodes as well as the incentive factor, energy trust and quality-of-service (QoS) indicators must be considered to improve the safe routing pathways. Therefore, a lightweight and fast deployable trust-based secure routing protocol (TBSRP) is developed to identify and isolate the misbehaving nodes to minimize the computational complexity of the nodes [38]. Additionally, the TBSRP employs a distributed trust mechanism to dynamically identify malicious nodes and isolate them as soon as feasible. The most trustworthy and quickest routing path is chosen based on the nodes’ trust level and hop count. However, it does not account for nodes’ energy levels when computing their trust levels, which might lead to choosing the next hop from nodes with high trust but low energy. The energy-efficient clustering approach suggested for an IoT network, is a revolutionary and adaptive energy-efficient fuzzy (AEEF) clustering that boosts efficiency by up to 10% and directly impacts longevity [39]. In this study, the author refined the energy distribution method and used fuzzy logic clustering to imitate human reasoning and cognition. Rather than strictly binary cases of
truth, fuzzy logic includes 0 and 1 as extreme cases of truth but with various intermediate degrees of truth, both of which have a direct impact on scalability. In particular, the AEEF algorithm presented a Landslide Detection System (LDS), which streamlines routing paths and finds the best routes via sinks to shorten delays and be useful in the event of landslides. The Reliable Cluster Head Selection Technique (RCHST) utilizes Integrated Energy and Trust-based Semi-Markov Prediction (IETSMP) to combine energy and trust parameters as RCHST-IETSMP [40]. This method leverages short-path communication using effective distance to carry out a basic comparison between the interacting sensor nodes of the IoT network and has the ability to aid the user in data processing and sensing. It takes advantage of the Hyper-Erlang process to lengthen a sensor network’s lifespan. The lifespan is a crucial element of an IoT network, whereas the Hyper-Erlang process requires more resources to extend the network’s lifetime. Dynamic directional routing (DDR) is designed for Mobile Wireless Sensor Networks (MWSNs) for effective and reliable routing that anticipates the next opportunity through advanced calculations of a fitness function [41]. Furthermore, it chooses the particular specified search angle that forecasts the total potential candidate for the following hope node. The discovery phase and the data forwarding phase are the two primary aspects of this protocol. It connects certain sophisticated characteristics of prediction with the fundamental Threshold sensitive Low Energy Adaptive Clustering Hierarchy (T-LEACH) structure. Due to the fact that T-LEACH chooses cluster heads at random which has an adverse effect on the algorithm’s overall efficiency, the DDR can only make limited improvements to its prediction, leading to low latency. The vulnerability and system weaknesses that are raised by the expansion of IoT networks have raised many concerns for the research community. Earlier, software-based intrusion detection systems (IDSs) were suggested as a solution to the security issues but over time, it became apparent that there is no clear and defined software methodology for IDSs. In order to extend the longevity and technology, the authors presented a Software Engineering (SE) (https://www.mdpi.com/1424-8220/20/5/1375, accessed on 14 August 2023) based solution for IoT in which three crucial SE phases are utilized as requirements, design and software testing to design the IDS [42]. When IDS is used for an IoT network, this approach performs well in terms of energy usage and network longevity. Since the broadcast design transmits many messages at once, it requires a lot of energy to do so, making it inefficient to implement. The key disadvantage of the suggested method is that complicated designs still require a correct structure to be solved. In order to attain the energy efficiency of IoT networks, the authors presented the Gaussian elimination approach with the Distributed Energy-Efficient Clustering DEEC Gaussian (DEEC Gaussian) in [43]. The Gaussian elimination algorithm is used to increase residual energy and improve optimal energy along with aggregation of IoT operations whereas the DEEC algorithm is used to provide a high-performance level of heterogeneity and achieve energy-efficient clustering that is used to increase the network lifetime in stable ways. Overall, none of the aforementioned encryption techniques, however, cannot successfully fight against assaults initiated from within the IoT network.

The paper introduces an Efficient and Secure Fog-Based Routing Mechanism (ES-FRM) to tackle the vulnerability of Internet of Things (IoT) networks to both internal and external attacks, as well as energy consumption challenges. Each IoT node calculates a comprehensive trust value for the next intermediate node, taking into account direct trust, indirect trust, and energy trust values before forwarding data. By comparing this trust value with a predefined threshold, the system identifies rogue nodes and ensures network protection.

3. Methodology

The proposed methodology is explained in conjunction with the system preliminaries of mathematical models that have been used to get the intended results. For the last several years, it is undeniable that the research aim in IoT is to find the safe, reliable and efficient transmission of data from source to destination. Hence, the proposed work goal is to
implement a trust-aware security mechanism to make the data transmission path secure, smooth and seamless. This is accomplished by anticipating the future behavior of the nodes by evaluating their history and excluding compromised nodes that can execute various malicious activities. The proposed methodology is applied in three stages as shown in Figure 1.

![Figure 1. A block diagram of an IoT network to implement the proposed methodology of Efficient and Secure Fog Based Routing.](image)

The proposed methodology is based on trust-aware security, which has been considered one of the primary requirements for the sensor nodes to take part in data transmission. The comprehensive trust value of every sensor node is calculated through the four parameters such as direct trust value, indirect trust value, energy trust value and volatilization factor. This comprehensive trust value is compared with the predefined threshold comprehensive trust value of every node and if the comprehensive trust value of any node is greater than or equal to the threshold comprehensive trust value, that node is declared as trustworthy otherwise it is notified as a malicious and faulty node. In this proposed paradigm, the IoT is thought to be made up of $n$ sensor nodes spread throughout the $m \times m$ square region of interest. Once every node collects data, it transmits to respective cluster head, after which it is passed to a sink node or BS via trustworthy neighbors. Aside from this, the proposed model makes the assumption that all the sensor nodes are supposed to be placed randomly across a $500 \times 500$ m² region of interest. All of the nodes are immovable, so once they are installed, they cannot be moved and each node in the IoT has its own unique ID. There are mathematical preliminaries are used to get the results of the proposed methodology which are discussed in the next sub-sections. The unit of trust value would be a dimensionless value expressed on a scale by combining the factors discussed in the above section, just like usage and remaining power of any node and many more. The scale is between 0 to 150 (watt unit of energy) in case of individual calculation and 0 to 300 watts in case of T-comp calculation.

3.1. Trust

The level of confidence that one node has in other nodes is known as the trust. Each node in the wireless sensor network has the ability to determine how trustworthy its neighbors are. Additionally, the term “neighboring nodes” refers to nodes that are situated...
within the transmission range of that specific node. Hence, calculating the trust value of every neighboring node is a prerequisite for taking part in data transmission, which can be changed over a period of time. Moreover, the node having a trust value higher than other nodes is most likely to be selected as the next data-relaying node.

3.2. Direct Trust

The method of gathering data for trust computation is deemed one of the most crucial elements of trust management schemes. The direct trust value of any node is computed by considering the numerous trust metrics of that particular node in various events of the network. The trust metrics, or the QoS attributes that can be considered are listed in as 0 to 1 [1]. The provided trust metric information for various events is crucial and can help the system’s trust management system for the node to make the right decisions. The direct trust is determined by the sending device itself while taking into account trust parameters. As a result, Equation (1) is used to determine the direct trust value of the nearby nodes so that every node can ascertain the direct trust value of each of its neighbors by using this equation.

$$T.\text{Direct}_{xy}^{n} = \theta \times (\alpha_{1}V_{t} + \alpha_{2}B_{t})^{n-1} + (1 - \theta) \times (V_{t} + B_{t})^{n}$$ (1)

The earlier portion of this equation illustrates the historical trust value, and the later part represents the current trust value. Moreover, the variables $\theta$ and $(1 - \theta)$ are weighting parameters and have been employed to adjust the current and past trust values. The $n$ is very important here which represents the number of nodes participating in an IoT network as the initial and current state of the network, whereas $x$ and $y$ show the coordinates and location of the nodes. In addition to this, the value of $\theta$ should be greater than 0 and less than 1 as per the specific application of the IoT network, whereas $\alpha_{1}$ and $\alpha_{2}$ are the volatilization factors and have been added to reduce the previous trust values of malicious nodes. Additionally, the symbols $V_{t}$ and $B_{t}$ represent the ratios of the total number of received and sent messages, respectively, out of all messages and are calculated using Equation (2).

$$V_{t} = \frac{\text{Get Message}_{y}}{\text{Total Message}_{y}}$$ (2)

Equation (2) is used to compute the ratio of received data packets out of all packets of sensor $y$, whereas Equation (3) is used to calculate the ratio of sent data packets out of all packets.

$$B_{t} = \frac{\text{Sent Message}_{y}}{\text{Total Message}_{y}}$$ (3)

3.3. Indirect Trust Value

The indirect trust of any node is computed by getting indirect information from the common neighboring nodes. So, Equation (4) has been used in the proposed methodology to calculate the indirect trust value of any node.

$$T.\text{Indirect}_{xy}^{n} = \frac{1}{p} \sum_{C_{i} \in Ch} \left( T.\text{Direct}_{xC_{i}}^{n} \times T.\text{Direct}_{Cy}^{n} \right)$$ (4)

As seen in Equation (4), $C_{h}$ is the complete set of shared neighbors that both the trust evaluator nodes and the nodes whose trust is being assessed. Here $p$ represents the total number of common neighbors. Moreover, in order to compute the indirect trust value from any node $x$ to other node $y$, first of all, direct trust is calculated from node $x$ to common neighbors $C_{x}$ and subsequently from common neighbors $C_{y}$ to node $y$, as illustrated in Equation (4). After calculating the common neighbor node’s direct trust value, it is compared to the threshold direct trust value. If their direct trust value is more than the threshold value or equal to it, it is proclaimed to be a trusted neighbor; otherwise, it is excluded from the list of common trusted neighboring nodes.
3.4. Volatilization Factor

The volatilization factors $\alpha_1$ and $\alpha_2$ have been added to the suggested work in order to quickly remove any sensor nodes that have been transformed from reliable working nodes to potentially dangerous nodes that could endanger the safety of the route, as shown in Equations (5) and (6).

$$\alpha = e^{-a_1 \mod (T, t)} \quad (5)$$

$$\alpha = e^{-a_2 \mod (T,t)} \quad (6)$$

The variables $a_1$ and $a_2$ have been applied to adjust the fluctuations in trust values of environmental factors for flat or geometry-based atmospheres. Additionally, $T$ represents the current time in the network, whereas $t$ stands for the threshold of time. While mod $(T, t)$ has been used to prevent the labeling of numerous lawful nodes that have been deployed in some distant areas but are not taking part in data transfer as malicious nodes.

3.5. Residual Energy

The residual energy $E_r$ of a sensor node is the energy that is spent during the transmission and reception of data through the network, and Equation (7) is used to determine the energy trust value of the node [3].

$$\text{Data Recieving Power } y = K \times E_{elec} \quad (7)$$

where $K$ is the total number of bits, and $E_{elec}$ is the electronic energy consumed to receive one bit of data. The total energy for sending the data of node $y$ is calculated using Equation (8).

$$\text{Data Sending Power } y = \begin{cases} K \times E_{elec} + K \times \epsilon_{fs} \times d^2 \text{ where } d < d^2 \\ K \times E_{elec} + K \times \epsilon_{mp} \times d^4 \text{ where } d < d_0 \end{cases} \quad (8)$$

Here, $d_0$ represents the threshold distance and calculated using Equation (9), $E_{elec}$ shows the energy utilization in the electronics for sending or receiving one bit. The terms $\epsilon_{fs} \times d^2$ and $\epsilon_{mp} \times d^4$ are amplifier energy consumptions for a short and long-distance transmissions.

$d_0$ is calculated using this formula

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \quad (9)$$

If it is supposed that the node $y$’s initial energy is $E_0$, then its remaining energy is computed via Equation (10).

$$\text{Remaining energy of the node } y = E_0 - \text{Data Recieving Power } y - \text{Data Sending Power } y \quad (10)$$

Finally, Equation (11) is used to calculate the energy trust value of each sensor device.

$$E_y = \frac{\text{Remaining energy of the node } y}{E_0} \quad (11)$$

After calculating energy trust value $E_y$, it can be deduced that this node can only take part in data transmission if its residual energy is more than or equal to the threshold energy of the node.

3.6. Comprehensive Trust Value

The comprehensive trust value of any node is calculated by adding direct trust, indirect trust and energy trust values of the node as given in Equation (12).

$$\text{Comprehensive Trust value}^{n}_{xy} = \omega_1 \times T.\text{Direct}^{n}_{xy} + \omega_2 \times T.\text{Indirect}^{n}_{xy} + \omega_3 \times E_y \quad (12)$$
Here the $\omega_1$, $\omega_2$ and $\omega_3$ are the weighting coefficients for direct trust, indirect trust and energy trust values, respectively, and $0 < \omega_1, \omega_2, \omega_3 < 1$, $\omega_1 + \omega_2 + \omega_3 = 1$.

### 3.7. Encryption of Data

A method that solves one of cryptography's most difficult problems is RSA encryption [2]. The creation of keys for data encryption is the first step in the RSA algorithm. It calls for the use of two prime numbers ($S$ and $T$), which are determined by primality testing. The prime numbers used in RSA must be extremely large and sufficiently spaced apart. Smaller or closer numbers are much more likely to break. Despite this, the process has been simplified by using smaller digits. $S$ and $T$ are the two prime numbers that are used as key factors, then multiplex factor is given by Equation (13).

$$R = S \times T$$  \hspace{1cm} (13)

Here $R$ is a multiplex key of prime factors that can be scaled using Equation (14).

$$\lambda(R) = \text{lcm}(S - 1, T - 1)$$ \hspace{1cm} (14)

where $\lambda(R)$ is the scaling factor that is used to make the key values stable. Now the encryption can be done using Equation (15).

$$E = \text{Plaintext}^e_m \mod R$$ \hspace{1cm} (15)

where $E$ represents the final data after encryption, which is known as Cipher text. Here $e$ is the estimated values of the plaintext normally given and $m$ is the mobility of the message that is also initialized at the start.

$$D = \frac{1}{e} \mod d_1$$ \hspace{1cm} (16)

Finally, the division matrix $D$ of message and data packages is calculated using Equation (16), where $d_1$ is the initial distance between the nodes. This whole mathematical process is shown in Figure 2 to find the appropriate cluster head sensor node, as per its security parameters. Algorithm 1 is used to calculate the overall trust value of an IoT node participating in an IoT network.

### Algorithm 1. The pseudocode for calculating Trust Value

**Input:** Node, Security, Energy  
**Output:** Trust value calculation and communication.  

Trust value of the node: $T_{\text{Value}} = (\text{Successful\_Trans} + \text{Unsuccessful\_Trans})/(\text{Trans} + \text{Successful\_Trans})$;  
if ($T_{\text{Value}} = $ adequate for communication); Allow communication with the node.  
else calculate the trust value for the security model of the node.  
T\_Security = Access\_Control + Encrypted\_Packets + Secure\_Routed\_Packets;  
if ($T_{\text{Security}} = $ adequate for communication); Allow communication with the node.  
else calculate the trust value for the energy consumption  
if ($T_{\text{Energy}} = $ adequate for communication); Allow communication with the node.  
else calculate the trust value of mobility. $T_{\text{Energy}} = \text{Mobility} + \text{Energy\_cons}$;  
if ($T_{\text{Mobility}} = $ adequate for the communication); Allow the communication.  
else calculate the trust for the node in the reliability model. T = SentData + Energy;  
if ($T = $ adequate for communication); Allow communication;  
else calculate the overall trust for the node.  
Overall\_Trust=$T_{\text{Value}} + T_{\text{Security}} + T_{\text{Energy}} + T_{\text{Mobility}}$;  
if (Overall\_Trust = adequate for communication); Allow the communication.  
else Deny communication with the node.  
end.
Figure 2. The mathematical implementation of Efficient and Secure Fog Based Routing in IoT network.

4. Results and Discussion

To check the performance of the proposed secure and efficient fog-based routing mechanism for IoT networks, it has to be understood that the data must not be intercepted by any hostile nodes as it travels from source to destination. Therefore, to ensure secure data communication for all participating nodes they must not carry out any attacks, such as black hole attacks, flood attacks, sinkhole attacks, etc. Energy preservation is also considered the most important factor for any IoT network during all this communication to enhance the network lifespan. For this purpose, the proposed technique implemented the clustering technique, which is to produce clusters aiming to save the energy of a sensor node. Hence, the data received by the intended node must be the same as the sent. This implies that there must not be any intrusion or alteration by any attackers in packets on their way to the destination. The proposed methodology was simulated using the parameters as shown in Table 1. The overall results are compared on the basis of comprehensive trust values, encoding and decoding of packets at three different levels such as from CH to Fog, Fog to Cloud and directly from CH to Cloud in the following sub-sections.

Table 1. The simulation parameters of Efficient and Secure Fog Based Routing in IoT network.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Node’s Initial Energy</td>
<td>0.87 J</td>
</tr>
<tr>
<td>Area of Interest (The concrete simulation scene)</td>
<td>500 m × 500 m</td>
</tr>
<tr>
<td>Data Length</td>
<td>200 bits</td>
</tr>
<tr>
<td>Threshold Comprehensive Trust Value [0, 1]</td>
<td>0.78</td>
</tr>
<tr>
<td>Malicious Nodes</td>
<td>Random</td>
</tr>
<tr>
<td>Probability of Clusters</td>
<td>0.034</td>
</tr>
</tbody>
</table>

4.1. Direct Trust, Indirect Trust and Comprehensive Trust

The direct trust value for every node denoted by T1 is computed using various availability metrics such as the node’s packet forwarding history, packet delay, residual energy, reputation, etc. Next, the indirect trust value T2 is derived using the recommendations of the shared neighbors, as shown in Figure 3. The direct trust and indirect trust values are first calculated and combined with weighting variables to provide a comprehensive trust value, which is represented as a T-comp against each node. Due to the drop in their
earlier trust values as well as in the ratio of delivered data packets because of volatilization factors, the comprehensive trust values of hostile nodes executing black hole attacks are significantly lowered, as shown in Figure 3.

![Figure 3. IoT Network of 100 Sensor Node.](image)

Additionally, due to the hello flood attack, the volatilization factor causes the value of received data packets of faulty nodes to become zero, which pushes the node’s overall comprehensive trust value below the threshold comprehensive trust values. Moreover, the comprehensive trust values of nodes launching selective forwarding and sinkhole attacks are also lowered than the threshold comprehensive trust value. As a result, the route would become safe and secure since all nodes whose comprehensive trust value appears below the threshold would be disqualified for data transfer.

4.2. Comparison of Comprehensive Trust Values

Figure 4 explains the comparison of the comprehensive trust value of the proposed Efficient and Secure Fog Based Routing Mechanism (ESFRM) and Trust-aware Secure Routing Protocol (TSRP) approach by putting the intended parameter values. The results show that the ESFRM is providing 16.44% better results than TSRP.

![Figure 4. Comparison of Comprehensive Trust Value.](image)

4.3. Encoding of Data from CH to Fog Nodes

The encoding of data when it travels from the cluster head to the fog node is shown in Figure 5. The data packets are encrypted so that no one can intercept them and compromise their integrity. The lower curve shows the values of the original data denoted by (xd), and the red upper curve denoted by (encode) shows the data when it has been encrypted.
4.3. Encoding of Data from CH to Fog Nodes

The encoding of data when it travels from the cluster head to the fog node is shown in Figure 5. The data packets are encrypted so that no one can intercept them and compromise their integrity. The lower curve shows the values of the original data denoted by \( (x_d) \), and the red upper curve denoted by \( \text{Encode} \) shows the data when it has been encrypted.

4.4. Decoding of Data at Fog End

When data reaches the fog node, Figure 6 shows how it is decoded. The decryption key is used to decode the data at the fog node, where it will be quickly processed before being sent to the cloud server. As can be seen, the original data values are the same even though the curves are displaced as a result of mixing the decryption key. The original content is the same as was sent by the source.

4.5. Encoding of Data from Fog Node to Cloud Server

Figure 7 shows how data is encrypted as it travels from a fog node to a cloud server. Data is transported to the cloud for further processing and long-term storage after having briefly processed here at the fog node because the cloud server is where the data will ultimately reside. The bottom curve represented by \( (y_d) \) in this figure shows the original decoded data values. When the data is encrypted using an encryption key, the original values are displaced as a result of the encryption key. Values of the encrypted data are represented by the upper curve.
4.5. Encoding of Data from Fog Node to Cloud Server

Figure 7 shows how data is encrypted as it travels from a fog node to a cloud server. Data is transported to the cloud for further processing and long-term storage after having briefly processed here at the fog node because the cloud server is where the data will ultimately reside. The bottom curve represented by $(y_d)$ in this figure shows the original decoded data values. When the data is encrypted using an encryption key, the original values are displaced as a result of the encryption key. Values of the encrypted data are represented by the upper curve.

4.6. Decoding of Data at Cloud Server

The decoding of data after it reaches the cloud server is described in Figure 8. The integrity of the data stays the same when it is decrypted using the decryption key at the cloud end, demonstrating that it was not altered on the way to the cloud server.

4.7. Data Encoding Directly from CH to Cloud Server

In contrast to the (SEFR) study, the encryption of data directly from cluster head to cloud server is also implemented in proposed paradigm. In Figure 9, $(z_d)$ with blue curve shows the original data at cluster head, and the red curve explains the encrypted data when the encryption key is applied.

---

**Figure 7.** Encoding from Fog Node to Cloud Server.

**Figure 8.** Decoding of Data at Cloud Server.

**Figure 9.** Data Encoding from CH to Cloud Server.
Figure 9. Data Encoding from CH to Cloud Server.

4.7. Data Decoding from CH to Cloud Server

Figure 10 illustrates how data is decoded when it is received straight from the CH to the cloud end. In this case, the data is thoroughly processed after being decrypted using the decryption key. Additionally, it can be seen that when the data is decrypted, the original content of the data is preserved, which serves as the primary protection against outsiders accessing the data’s original values.

Figure 10. Decoding of Data for Direct Transmission at Cloud End.

4.8. Data Decoding from CH to Cloud Server

Figure 10 illustrates how data is decoded when it is received straight from the CH to the cloud end. In this case, the data is thoroughly processed after being decrypted using the decryption key. Additionally, it can be seen that when the data is decrypted, the original content of the data is preserved, which serves as the primary protection against outsiders accessing the data’s original values.

Figure 11 shows how much energy is utilized when data is transferred straight from the CH to the fog node in S1 and the energy used from the fog nodes to the cloud server in S2. The results explain that communication of data from CH to the fog node consumed more energy because of no processing, filtering and refining of data here at the initial stage after being sensed. Additionally, when data reaches a fog node, it is refined and processed, appropriate computations are conducted, and operations are prompted here rather than transmitting the complete data set to the cloud server. Because of this, only the data that is not currently needed is transferred to the cloud server for processing and storage. As a result, sending data from a fog node to the cloud server, represented by S2 in the above diagram, uses less energy on average.
4.10. Energy Consumption for Direct Transmission from CH to Cloud Server

Figure 12 shows how much energy is utilized when data is transferred straight from a CH to a cloud server without the use of any between nodes. The idea is that data is sent directly to the cloud server in this case rather than the fog node, which, on average, uses more energy as shown in the figure. This is because it is obvious that without an intermediate relaying node, more energy from the source node is used to send the data to the cloud server.

4.11. Energy Consumption Comparison between Proposed and Base Study

The energy usage comparison between the suggested paradigm (ESFRM) and the (SEFR) is shown in Figure 13. As a result, it is demonstrated that the proposed study used an average of 13.67% less energy than that of the based technique when applying intended parameter values.

4.12. 2-Way and Direct Transmission Rate of Nodes

Figure 14 shows the data transmission rate of all the network participants from the CH to fog, the fog node to the cloud server, as well as directly from CH to the cloud server. The average rate is more effective in two-way communication than in direct transmission, as can be observed because in two-way transmission, data is not intended to go as far as it does in direct transmission. Data is transmitted from the CH to the fog node, and then from the fog node to the cloud server. We receive a better transmission rate as a result.
4.11. Energy Consumption Comparison between Proposed and Base Study

Figure 13. Energy Consumption Comparison between Proposed and Base Study.

4.12. Transmission Rate of Nodes for Two Way Transmission and DT

Figure 14. Transmission Rate of Nodes for Two Way Transmission and DT.

4.13. Overall Threshold Comparison between ESRM and TSRP

The comparison of the overall threshold values between the suggested and the base research is shown in Figure 15. These threshold values indicate how many data packets are transferred in a given amount of time. The proposed technique is showing results that are 16% better than the TSRP as the number of nodes grows.

Figure 15. Transmission Rate of Nodes for Two Way Transmission and DT.
The comprehensive trust value of each node is evaluated and compared against a predefined threshold to determine whether it should be categorized as a rogue node or a suitable normal node. To enhance data security during transmission, an advanced RSA algorithm is employed, safeguarding data as it moves from the Cluster Head (CH) to fog nodes, from fog nodes to the cloud server, and even directly from CH to the cloud server. Additionally, the protocol utilizes a clustering technique to ensure efficient Cluster Head selection. Extensive simulations demonstrate that the proposed Enhanced Secure Fog-based Routing Mechanism (ESFRM) surpasses the performance of the Trust-Based Secure Routing Protocol (TSRP) by achieving a 16.44% higher malicious nodes detection rate. Furthermore, ESFRM exhibits a remarkable energy efficiency advantage, consuming approximately 13.67% less energy compared to the Secure Energy-efficient Fog-based Routing (SEFR) protocol.

5. Conclusions

Due to the uncertain and rapidly changing behavior of the sensor nodes, energy-efficient and reliable routing remained a great matter of concern in resource-constrained-fog-based IoT networks. Therefore, this paper proposes Energy Efficient and Secure Fog Based Routing Mechanism (ESFRM) to defend against various internal as well as external attacks. The main purpose of ESFRM is to identify the malicious and faulty internal nodes as soon as possible to prevent them from disrupting the normal transmission of data from source to destination node. For this purpose, comprehensive trust value is computed which encompasses the volatilization factor in addition to direct trust, indirect trust and energy trust values. The comprehensive trust value of every node is compared with the threshold comprehensive trust value to decide whether it is a rogue node or an appropriate normal node. Moreover, an enhanced RSA algorithm has been utilized to prevent unauthorized access when data is traveled from CH to fog node, fog node to cloud server and directly from CH to cloud server in addition to clustering technique for the sake of efficient CH selection. The simulation results show that the malicious nodes detection rate of the proposed ESFRM is 16.44% higher than TSRP and it utilizes, on average, 13.67% less energy than SEFR. In the future, the proposed methodology can be implemented and extended within the IDS to secure the IoT network from internal as well as external attacks.

Author Contributions: Conceptualization, T.S.M. and J.T.; Data curation, M.R.M.; Formal analysis, S.A. and M.R.M.; Funding acquisition, A.K.; Investigation, S.A.; Methodology, T.S.M., J.T. and H.A.; Supervision, A.K.; Writing—original draft, T.S.M., S.A. and H.A.; Writing—review & editing, M.R.M. and A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Taherdoost, H. Security and Internet of Things: Benefits, Challenges, and Future Perspectives. Electronics 2023, 12, 1901. [CrossRef]


**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.