Received: 1 June 2019 | Revised: 6 September 2019 | Accepted: 11 January 2020

DOI: 10.1002/dac.4341

RESEARCH ARTICLE

WILEY

# Trust-based secure routing and the prevention of vampire attack in wireless ad hoc sensor network

R. Isaac Sajan[1] | J. Jasper[2]

[1]Department of Computer Science and Engineering, Ponjesly College of Engineering, Nagercoil, India

[2]Department of Electrical and Electronics Engineering, Ponjesly College of Engineering, Nagercoil, India

**Correspondence**
R. Isaac Sajan, Department of Computer Science and Engineering, Ponjesly College of Engineering, Nagercoil, India.
Email: isaacsajanr.001@gmail.com

**Summary**

The recent development in technologies is the driving force for development in sensors, especially in military applications. Due to the openness nature of the ad hoc sensor network, the system gets easily affected, which may lead to some packet drop, transmission delay, high network overhead, and more energy consumption. In this paper, the security during data transmission is provided to the network by using the proposed Secure Atom Search Routing (SASR) algorithm, which is adopted from the behavior of molecular dynamics. For global optimization problems, this algorithm provides an effective solution based on the constraint and interaction force of atoms. Moreover, the performance of SASR is improved by providing a proper balance between exploitation and exploration. Since the knowledge base processes all the data, the computational complexity is reduced and the lifetime of the network is increased. The simulation and performance are carried out for the proposed Knowledge and Intrusion Detection based Secure Atom Search Routing (KID-SASR) protocol and is compared with the existing methods based on the metrics trust, delay, throughput, energy, packet delivery ratio, network lifetime, trust detection rate, and communication cost. The results obtained show improvement in the overall performance of the system.

**KEYWORDS**

atom search routing, carousal attack, intrusion detection, stretch attack, trust, vampire attack, wireless ad hoc sensor network

## 1 | INTRODUCTION

A wireless network is a way of communication between various nodes such as a computer, mobile phones, printer, and other devices through physical infrastructure like router.[1-3] Due to technological development, ad hoc sensor networks are being implemented in various sectors like the military, health care, and defense. Ad hoc network does not require any physical infrastructure like routes; instead, every node connected to the network act as a router. These sensors are used in monitoring environmental conditions like humidity, temperature, movement, traffic monitoring, noise, military applications, and agriculture farm management. Since these sensors are battery-operated devices, they are vulnerable to Denial of Service (DoS),[4-6] which is caused by malicious attacks such as vampire attack, black hole attack, directional attack, and selective forwarding attack. The prevention techniques of these attacks are studied in Gunasekaran and Periakaruppan [7] and Osanaiye et al.[8] Vampire attacks are not protocoled specific, hence difficult to detect.

The vampire attack can be classified based on the stateless protocol and state full protocol.[5] In the stateless protocol, the entire routing path to the destination is specified by the source node itself. In the state full protocol, each node makes independent or self-decision to find the optimal routing path such as Optimal Link State Routing (OLSR)[9] and Destination Sequenced Distance Vector (DSDV) protocol.[10] The two types of attacks that can occur in a stateless protocol are carousal and stretch attack. The carousal attack is a type of attack that causes the data packet to send in a loop, not allowing the data to reach the destination. This causes drainage of battery resulting in DoS. Stretch attack occurring in stateless protocol causes the data packet to take a longer route to reach the destination resulting in more energy consumption.

The vampire attacks are not dependent on faults during implementation or design properties. The adversaries do not have the intention to tamper or forge the data packet but to completely drain their energy by gradually sending useless packets. The cyrptographic techniques may be helpful to prevent tampering of data but does not provide prevention against carousal and stretch attack, which aims to reduce the battery life. They not only flood the network with some useless message but gradually send some data packet to a node in order to completely deplete their energy. Moreover, if the sensor node is a foreign node that is not a part of the network, the cryptographic techniques are efficient in terms of providing strong security but if the sensor node is compromised by the attacker, then the security of the network during data transmission is exposed.

This paper is discussed and organized as follows: Section 2 gives the literature review. In Section 3, the proposed Knowledge and Intrusion Detection based Secure Atom Search Routing (KID-SASR) protocol is explained with trust verification of the nodes. In Section 4, the secure routing algorithm proposed is explained and is pictured with a suitable architecture diagram and flowchart. In Section 5, the performance metrics taken are formulated. In Section 6, the simulation setup, parameter setting, routing model, and comparative analysis are provided, and finally, Section 7 provides the conclusion of the paper.

## 2 | LITERATURE REVIEW

This section provides a review of existing protocols that are intended to secure the network against attacks and their drawbacks.

Various techniques and protocols are implemented to prevent these attacks such as Ariadne, which is an on-demand routing protocol, Low Energy Adaptive Clustering Hierarchy (LEACH) protocol, which is developed for reducing battery consumption.[11] LEACH protocol has a steady-state phase and a setup phase. In the steady-state phase, the cluster is formed based on the threshold value, where each node calculates its value by selecting a random number from 0 to 1 and broadcasts the value to its neighbors. For a node whose value is below the threshold limit will be nominated as the cluster head (CH) and remaining nodes will join as a member node. LEACH protocol helps in reducing energy consumption but does not provide secure routing against the vampire attack.

Energy Weighted Monitoring Algorithm (EWMA) has two phases[12,13]: the network configuration phase and the communication phase. In the network configuration phase, it provides an optimal routing path from the source node to the destination node, and in the communication phase, the duplicate data packets are avoided. The cryptographic keys can protect against outside attacks called active attacks but are not reliable in case of passive attacks that affect quality of service and reliability.

To avoid the passive attack and for the optimal selection of the path, optimization algorithms such as the Lion algorithm[14,15] were introduced based on the lion's behavior during territorial defense and takeover. The solution for nonlinear system identification is carried out. But solutions for complex problems are not considered. Hence, whale optimization algorithm (WOA), which is also an optimization algorithm, was proposed by Seyedali Mirjalili and Andrew Lewis[16] by absorbing the behavior of humpback whales. Since the location of the prey is not known initially, the current solution or prey is considered as the best solution and the position is updated in each iteration, but the nodes in the network consume more energy for computation.

Parno, Luk, Gaustad, and Perrig (PLGP) protocol has two phases. In the topology discovery phase, the address of each node in the network is learned, and in the packet forwarding phase, the next hop is decided by forming a virtual address within the nodes. PLGP protocol does not find whether it is moving closer to the destination during each hop. To overcome this disadvantage, the PLGPA protocol was introduced in Vasserman and Hopper,[5] where it checks if the data packet is moving closer to the destination during each hop by creating a virtual binary address for its cluster and its nodes. Creating a virtual address for each hop is not an efficient method since it causes delays and more energy consumption.

PROVEnance-based Trust model (PROVEST) was introduced by Cho and Chen,[17] where the route dynamically changes and in the presence of the malicious nodes, the trust of the node is evaluated. According to the Trust-Based Adaptive Acknowledgment (TRAACK) protocol proposed by Rajeshkumar and Valluvan,[18] acknowledgment is initiated on the chosen packets based on the trust value of the entire route. Here the intermediate nodes forward the packet to the destination. Once the packet is delivered, the destination node sends the ACK message to the source along the same route in reverse. This decreases the packet overhead, but the malicious discovery is not so reliable.

According to the AF-TNS algorithm[19] proposed by AlFarraj et al, two phases are classified: direct trust evaluation phase and addictive metric evaluation phase. Indirect trust evaluation phase, based on packet forwarded by the neighboring nodes, the trust is evaluated and in the addictive metric evaluation phase, identification and rectification of trusted path are carried out. But for complex calculations, overload on nodes occurs. EATSRA algorithm proposed by Selvi et al[20] has two phases, namely, trust-based secure routing phase and decision tree-based best path selection phase. In the first phase, a secure path is established among nodes having high trust, and in the second phase, routing is performed for the trusted node. However, low throughput and packet delay occur during transmission.

Jiang et al proposed EDTM protocol[21] to ensure trust for the network by calculating direct trust and recommendation trust. Direct trust can be obtained by finding out the communication, energy, and data. Recommendation trust is obtained from reliability and familiarity. This protocol is resistant against attacks but defined threshold, and weight is still the main challenge. In Ahmed et al[22], Bakar et al proposed TERP protocol with two phases. The node's trustworthiness is found out in the trust estimation phase and routing among trusted node is carried out in the trust-energy aware routing phase. It offers high throughput, minimum delay, and reduced routing load, but it is not an efficient method for vampire attacks.

In Labraoui et al,[23] the trustworthiness is determined by two methods: reputation evaluation and risk evaluation using RaRTrust protocol. This protocol effectively deals with malicious nodes. According to HEED protocol,[24] a new clustering scheme was developed by Younis and Fahmy depending on the weight of the node and local information, which prolongs the lifetime, but it is suitable for low-level hierarchy. Das and Islam proposed STRM[25] protocol by evaluating direct trust, historical trust, recent trust, and indirect trust, and finally, the expected trust is evaluated. This protocol proves to be more robust and more effective against malicious agents.

From the above analysis, secure transmission of data in a wireless system is processed to protect the system from the several attacks and the routing path established from the trust metrics of the protocol. The existing approaches consume more energy and lack in the routing of packets in an efficient manner. Also, due to the resource depletion attack, the delivery of messages is not in a trusted form; hence, node trust is essential for the dynamically changing environment. In order to mitigate these drawbacks, trust and intrusion detection system based secure transmission routing mechanism is needed to transmit the packet based on their trustworthiness of the node and an efficient path must be selected with the minimal node energy consumption.

This paper proposes KID-SASR protocol for monitoring the malicious activity of the network and provides a secure routing scheme based on atom search optimization algorithm in Zhao et al.[26] The intrusion detection system monitors the malicious activity in the network and sends it to the knowledge base (KB) placed in the base station (BS). Since the KB does all the complex computation, the overload on the nodes is eliminated. Finding trusted shortest path to the destination is also the main factor to be considered when it comes to vampire attacks. Hence, we introduce a Secure Atom Search Routing (SASR) algorithm where we update the position, distance, energy, and trust values of all the nodes in each iteration. Frequent position updating increases more reliability during data transmission, and the KID system reduces the overall energy consumption of the network. Table 1 provides the list of symbols with its description.

## 3 | KNOWLEDGE AND INTRUSION DETECTION BASED SECURE ATOM SEARCH ROUTING (KID-SASR) PROTOCOL

The stages involved in the proposed protocol for secure data transmissions in wireless ad hoc sensor involve the following stages: (a) cluster formation, (b) trust node verification, (c) malicious node detection, and (d) optimal route selection.

**TABLE 1** List of symbols

| Symbol | Meaning | Symbol | Meaning |
|---|---|---|---|
| $T(n)$ | Threshold of node "n" | $x_{i(t)}$ | Initial position of the node |
| $P$ | Percentage of the sensor node | $x_j(t)$ | Updated position of the best neighboring node |
| $R$ | Current round | $x_{best}(t)$ | Best position of the node |
| $S$ | Nodes other than CH | $m_{i(t)}$ | Mass (fitness) of node "i" |
| $E_{\text{cur}}$ | Current energy of the node | $a(t)$ | Acceleration of the node |
| $E_{\text{int}}$ | Initial energy of the node | $F_i(t)$ | Interaction force |
| $T^d(t)$ | Direct trust degree calculated between two nodes | $G_i(t)$ | Constraint force |
| $P_{n1}(t)$ | Packets received by node n1 | $\alpha$ | Depth weight |
| $P_{n2(t)}$ | Packets sent by node n2 | $\beta$ | multiplier weight |
| $T^I(t)$ | Indirect trust degree calculated by neighboring nodes | $r_{ij}(t)$ | Distance between the best neighboring node and the source node |
| $k$ | Number of neighboring nodes | $V_i(t)$ | Velocity of nodes |
| $E$ | Total energy consumed by the | $\text{P}_\text{r}$ | Number of packets successfully received |
| $D$ | Delay in packet delivery | $P_s$ | Number of packets successfully sent |
| $d(i,j)$ | Distance between node i and j | AT | Arrival time of packets |
| $f_{\text{best}}$ | Best fitness value | ST | Sent time of packets |
| $f_{\text{worst}}$ | Worst fitness value | $E_{elec}$ | Energy spent to run the transmitter |
| $\text{fit}_\text{i}(\text{t})$ | Fitness of $i$th node | $\varepsilon_{amp}$ | Dissipated energy in the transmit amplifier |
| $t$ | Current iteration | $\tau$ | Channel path loss exponent |
| $T$ | Maximum number of iteration | $E_R$ | The receiver energy consumed |
| $\sigma(t)$ | Distance between the source node and its best neighboring node | B | Number of bits |
| $N$ | Total number of sensor nodes | $C_{d(t)}$ | Cost of packets to be delivered |

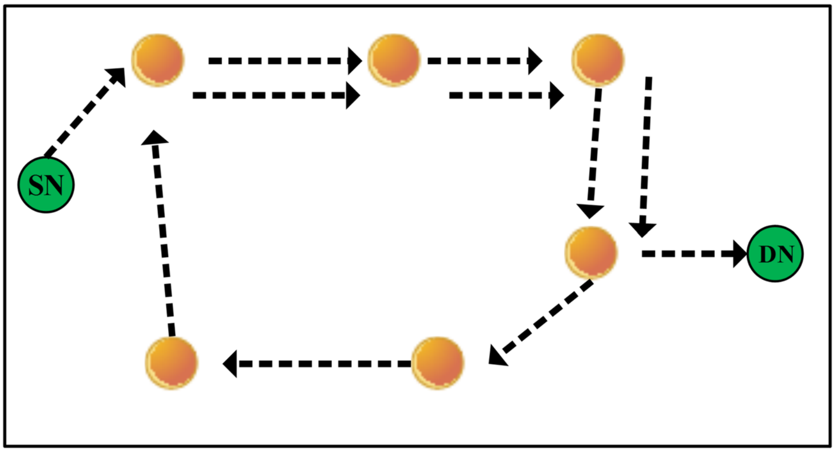
## 3.1 | Cluster formation

Once the sensor nodes are positioned in the network, each node calculates its energy from Equation (1) by selecting a random number ranging from 0 to 1 and broadcast the value to their neighboring nodes along with their ID. The minimum threshold is set to 0.05. In the next process, the nodes compare their threshold value with the threshold of the neighboring nodes. If the value is below the threshold of all other neighboring nodes, it nominates itself as CH and broadcasts the message to the neighboring nodes or else it joins as a member node with the sensor having minimum threshold value. In this way, the network is sectioned into several clusters with each cluster having a CH. These CHs are responsible for making a connection and sharing data with the neighboring clusters. The threshold of node "n" is determined by using Equation (1).

$$T(n) = \begin{cases} \dfrac{P}{1-P*\left(r\text{mod}\dfrac{1}{p}\right)} \times \dfrac{E_{cur}}{E_{\text{int}}} & \text{if } n \in S \\ 0 & \text{Otherwise} \end{cases}. \tag{1}$$

In the above equation; $E_{\text{cur}}$ is current energy of the node; $E_{\text{int}}$ is the initial energy of the node; $P$ is the percentage of the sensor node, which is the probability of a sensor node to become a CH; and $r$ is the current round. $S$ denotes the number of nodes other than CH. The nodes broadcast their calculated energy to all the neighboring nodes. The nodes, which have a value below the threshold, will be nominated or elected as a CH and other nodes join the CH as the member nodes.

**FIGURE 2** Carousal attack



## 3.2 | Knowledgebase and intrusion detection system

The intrusion detection system (IDS) monitors the malicious activity in the network by using a KB and inference engine. The KB situated in the BS stores the data collected from the CHs in the network, which is accessed by CH using the inference engine. Here, the inference engine is responsible for generating rules for the KB to perform the operation. Each member node generates events, which contain behavioral data. The CH monitors the activity of the node and shares the collected data with the BS where the data are verified. On verification, if a malicious node (nodes with low trust degree) is detected, which is found out by evaluating the fitness function provided in the below section 3.3, the IDS sends a warning message along with the data regarding the malicious node to the respective CH. Since the IDS performs all the operation, the computational complexity in the sensor node is reduced resulting in faster execution. The architecture of the IDS is depicted in Figure 1.

## 3.3 | Trust verification

In this section, trust refers to the degree of believability among nodes to transmit the data more securely. Nodes in the network are not only meant to deliver the exact data but it also updates additional information in the received data packet and sends it to the next node. Hence, it is very essential to ensure that the data sent between the nodes are confidential. A node becomes trustless when it is attacked by an active or passive attack. Two common attacks that can occur during data transmission are carousal attack and stretch attack. Carousal attack occurs when a malicious node makes the data packet to repeatedly circle in loops, not allowing the packet to reach the BS or the destination node. This abnormal behavior causes the energy of the node to deplete drastically within a short period resulting in DoS.

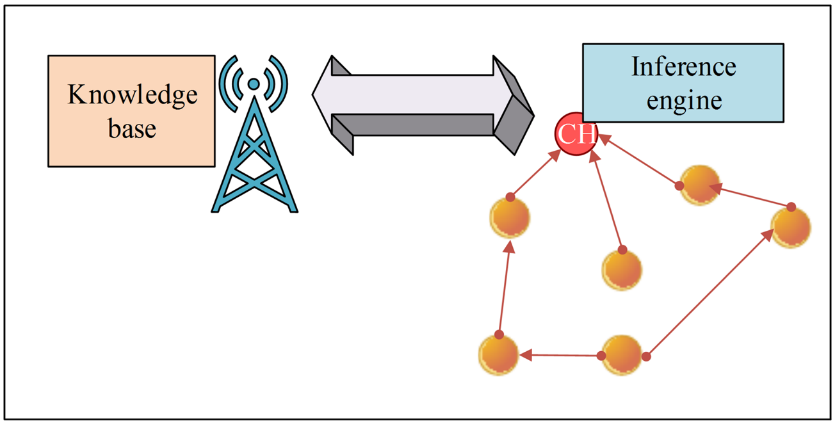


**FIGURE 1** Intrusion detection system

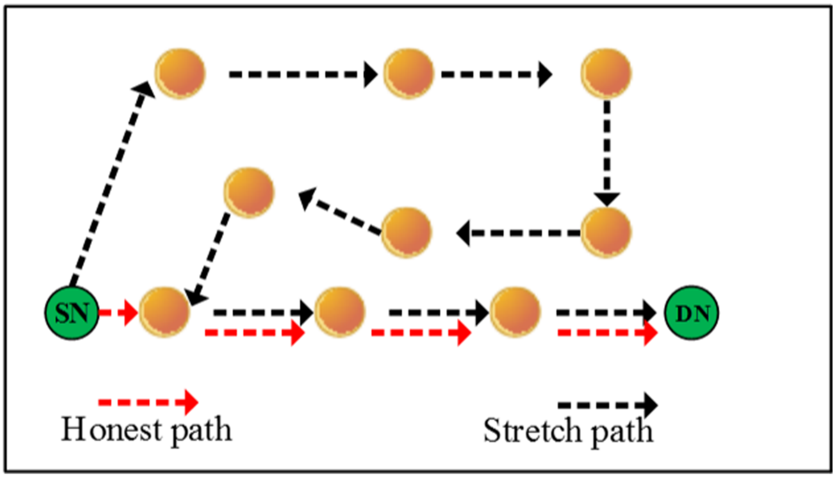


**FIGURE 3** Stretch attack

Figure 2 shows how the data are transmitted in a loop during the vampire attack. This loop may continue for two or more times. The stretch attack is another type of vampire attack that causes the data packet to take a longer route which ultimately depletes the battery. Figure 3 shows how the stretch attack affects among the nodes in the sensor networks, SN represents the source node. and DN represents the destination node.

Trust can be of two types: direct trust and indirect trust. Based on the interaction of two individual nodes, direct trust is evaluated. Indirect trust is calculated based on information from other neighboring trusted nodes. If the data packet requested is not received by the BS within the given time interval, the BS asks the CH to calculate the trust value of all its member nodes. The equation for direct trust is given by

$$T^d(t) = \frac{P_{n1}(t)}{P_{n2(t)}}. \tag{2}$$

Here $T^d(t)$ is the direct trust calculated between node n1 and n2.$P_{n1}(t)$ represents received packets. $P_{n2(t)}$ is the total packets sent. The trust calculated by the neighboring nodes (indirect trust) is given by

$$T^I(t) = \frac{1}{k}\sum_{d=1}^{k} T^d(t), \tag{3}$$

where $k$ is the number of neighboring nodes and $T^d(t)$ is the direct trust degree calculated by the neighboring nodes. Therefore, the total trust degree,

$$T = \alpha T^d(t) + \beta T^I(t). \tag{4}$$

$\alpha$, $\beta$ value ranges from 0 to 1 such that $\alpha+\beta = 1$. The threshold for trust is set to 0.99–1. If the calculated value drops beyond this value, then the sensor is considered as a malicious node since it has more packet loss during transmission. The trust factors are considered for the successful transmission of data to deliver the packet through the specified path.

## 3.4 | Malicious discovery

After the verification process, if the BS finds a malicious node based on trust degree calculation, it sends an alert message to the CH along with its ID and location of the discovered malicious node. The CHs now blacklist and isolate the malicious node and broadcast its ID to all its member nodes.

## 4 | SASR ALGORITHM FOR OPTIMAL ROUTE SELECTION

We introduce a SASR algorithm to find the best optimal route. Every substance around us is made up of molecules, and each molecule consists of atoms that are in motion in all the three states: solid, liquid, and gas. Atoms in a molecule make a bond with each other according to the distance between them. In molecular dynamics,[26] a set of atoms are randomly initialized. The position, velocity, and acceleration of atoms are updated in each iteration. On each iteration, the atom connects a bond with the best neighboring atom found so far based on the interaction and constraint forces. The same technique is implemented to find the best node in the WSN during attacks.

### 4.1 | Problem formulation for secure routing

The BS periodically collects the data regarding all the neighboring nodes of source node "*i*." From the collected set of data, the BS evaluates the fitness *f*(*x*) for the "*n*" number of neighboring nodes using the proposed SASR algorithm. After evaluation, the BS sends the ID of the best-identified neighbor to the source node "*i*." The objective function is executed for each position of nodes using the formula,

$$f(x) = \frac{1}{4}[T + E + D + d(i,j)]. \tag{5}$$

$T$ is the trust, $E$ is the total energy consumed by the node, $D$ is the delay in packet delivery, and $d(i,j)$ is the distance between node $i$ and $j$. Here, the trust is calculated from Equation (4).

#### 4.1.1 | Energy

The energy consumed by the sensor during data transmission should be maintained minimum. The consumed energy during transmission is given by

$$E_T = \left(E_{\text{elec}} + \varepsilon_{\text{amp}} d^{\tau}\right) B, \tag{6}$$

$E_{\text{elec}}$ is the energy spent to run the transmitter, $\varepsilon_{\text{amp}}$ is the dissipated energy in the transmit amplifier, $d$ is the distance of the receiver from the transmitter, $\tau$ represents channel path loss exponent ($2 \le \tau \le 4$), and $B$ represents total bits. The receiver energy consumed is given by

$$E_{R=} E_{\text{elec}} B. \tag{7}$$

The total energy consumed can be obtained by adding the energy consumption during transmission and reception.

#### 4.1.2 | Distance

Consider the position of node $i$ is taken as ($x_{11}$, $y_{12}$, $z_{13}$, $w_{14}$) and position of node $j$ is taken as($x_{21}$, $y_{22}$, $z_{23}$, $w_{24}$). Now the distance between two nodes $i$ and $j$ can be calculated by

$$r_{ij}(t) = \left\| x_j - x_i \right\| = \sqrt{(x_{21} - x_{11})^2 + (y_{22} - y_{12})^2 + (z_{23} - z_{13})^{(2)} + (w_{24} - w_{14})^2}. \tag{8}$$

### 4.1.3 | End-to-end delay

It is the excess time taken for the data packet to reach the sink. Vampire attacks like the stretch and carousal attack cause a high delay in packet transmission.

$$D = \frac{\mathrm{AT} - \mathrm{ST}}{K}. \tag{9}$$

AT is the arrival time, ST is the sent time, and $K$ is the number of connections.

## 4.2 | Routing design using SASR

Figure 4 illustrates an example of trust-based secure routing in the wireless ad hoc sensor network. In the above figure, the network is divided into different clusters (C=1, 2, 3) with each cluster is assigned with the CH (CH=1, 2, 3) and the member nodes of each cluster are represented as N, S, and M. The behavioral data are collected based on the information such as ID, energy of the nodes, distance, trust, delay, and address of source and destination and is sent to BS to find out any kind of malicious activity. Here, the node (N3, S2, M1, and M4) is identified as the malicious nodes in the network system. During data transmission, the routing path is established for forwarding the packets. At this phase, a secured path is selected by using the atom search routing algorithm to transmit the data in an efficient way with less energy consumption and discard the malicious packets. Here, the path (N1-N4-N5-CH1-CH2-M4-M7) is observed to have minimum distance. But due to the malicious node identification, the new route (N1-N4-N5-CH1-CH2-M6-M3-M7) is determined for data transmission.

### 4.2.1 | Initialization

SASR algorithm begins with the initializing phase where the number of nodes and their velocity is initialized.

### 4.2.2 | Fitness evaluation

$$\text{Minimize}, f(x), x = \left(x^{1}, .., x^{D}\right). \tag{10}$$

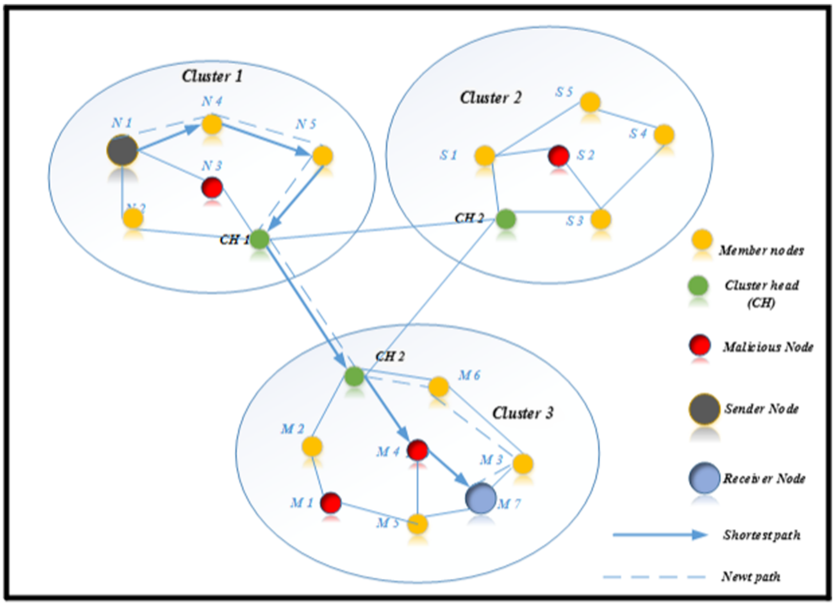


**FIGURE 4** Example of the proposed secure system model

The initial position of the *i*th node is expressed as

$$f(x_i), x_{i=}\left(x_i^1, \ldots, x_i^D\right) \tag{11}$$

*D* is the dimension of the matrix. Here we are considering four dimensions based on energy, delay, trust, and distance; *i* is the number of nodes. For example, the position of node 1 can be written as $x_{iD} = x_{1D} = (x_{11}, y_{12}, z_{13}, w_{14})$, the position of node 2 can be written as$x_{2D} = (x_{21}, y_{22}, z_{23}, w_{24})$, and so on. The fitness function is determined by using the Equation (5) given in section 4.1. After finding the fitness for all the nodes, the least value is considered as the best fitness value, $f_{\text{best}}$, and fitness with high value is considered as the worst fitness value, $f_{\text{worst}}$. In molecular dynamics, the masses of all the neighboring nodes are calculated. The atom with a larger mass will have the best fitness value, and the atom with lighter mass will have the worst fitness value. For the proposed system, the routing is selected based on the best path found so far,$f_{\text{best}}$ from the condition$S \le f_{\text{best}} \le D$, where *S* represents the minimum value of delay and distance $f_{\min}(D, d(i,j))$ and *D* represents the maximum value of trust and energy $f_{\max}(T,E)$. The nodes that satisfy the condition rearrange their mass values in descending order.

$$m_{i(t)} = \frac{M_i(t)}{\sum\limits_{j=1}^{N} M_j(t)}, \tag{12}$$

$$M_i(t) = e^{-\left[\frac{fit_i(t) - f_{\text{best}}(t)}{f_{\text{worst(t)}} - f_{best}}\right]}, \tag{13}$$

where $M_i(t)$ and $M_j(t)$ denote the mass of *i*th and *j*th atoms (node), respectively; *N* denotes the number of nodes; is the fitness of the *i*th node; $\mathrm{f}_{\text{best}}(\mathrm{t})$ is the minimum fitness value; and $f_{\text{worst}(t)}$ is the maximum fitness value.

Atoms interact with *K* neighboring atom to enhance the exploration. Similarly, the number of neighboring nodes in the sensor network at $t^{th}$ iteration is calculated as

$$K(t) = N - (N-2) \times \sqrt{\frac{t}{T}}, \tag{14}$$

where *N* is the total number of sensors, *t* is the current iteration, and *T* represents the highest iteration. *K* decreases gradually with each iteration in which the first five nodes that are close to the maximum fitness value are selected as $K_{\text{best}}$ neighboring nodes. The position is then updated for these best neighboring nodes as described in Section 4.2.3.

### 4.2.3 | Position updating

After finding the neighboring nodes from Equation (14), the mass rearranged in descending order is compared with the previous mass values and the new positions of nodes are updated as$x_{j=}\left(x_j^1, \ldots, x_j^D\right)$, where *j* represents the nodes, dimension *D*=4. Atoms bond with each other based on the Lennard-Jones (L-J) potential and bond length potential. Interaction force results from L-J potential, which is a mathematical model for the interaction between two atoms or molecules and constraint force results from the bond length potential. At the last stage, the atoms interact only with the atoms having the best fitness value $k_{\text{best}}$ to enhance exploitation. In the proposed SASR algorithm, the interaction force between two nodes is determined based on the distance between the source node and the neighboring $k_{\text{best}}$ nodes. The number of $k_{\text{best}}$ nodes decreases gradually with each lapse of iteration.

$$\sigma(t) = \left\| x_{ij}(t), \frac{\sum\limits_{j \in K\text{best}} x_{ij}(t)}{K(t)} \right\|_2. \tag{15}$$

In the above equation, $\sigma(t)$ represents the distance between the source node and its best neighboring node. For short distance, the interaction force will be high, and for long distance, the interaction force of the nodes will be low. The constraint forces of the nodes are calculated by finding the difference between the best position $x_{\text{best}}(t)$ and the initial position $x_{i(t)}$ of the node. Now, the acceleration of the node is calculated as

$$a(t)=\frac{1}{m_i(t)}\left[F_i(t)+G_i(t)\right]. \tag{16}$$

Equation (16) can be rewritten as

$$a(t)=\frac{1}{m_i(t)}\begin{bmatrix}-\alpha\left(1-\frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}\sum\limits_{j\in Kbest} rand_j\left[2\times\left(h_{ij}(t)\right)^{13}-\left(h_{ij}\right)^{7}\right]\frac{\left(x_j(t)-x_i(t)\right)}{\left\|x_i(t),x_j(t)\right\|_2}} \\ +\beta e^{-\frac{20t}{T}}\left[x_{best}(t)-x_{i(t)}\right]\end{bmatrix} \tag{17}$$

$m_i(t)$ is the mass obtained from Equation (12), $F_i(t)$ is the interaction force, $G_i(t)$ is the constraint force, $t$ represents the current iteration, $T$ represents the maximum number of iteration, $x_i(t)$ is considered as the position of the source node, $x_j(t)$ is considered as the updated position of the best neighboring node, $\alpha$ is depth weight taken as 50, and $\beta$ is the multiplier weight taken as 0.2. From the above Equation (17), it is estimated that the nodes with a larger mass will have low acceleration and the nodes with lighter mass will have high acceleration.

$$h_{ij}(t)=\begin{cases} h_{\min}\ , \dfrac{r_{ij}(t)}{\sigma(t)}<h_{\min} \\ h_{\max}\ ,\ \dfrac{r_{ij}(t)}{\sigma(t)}>h_{\max} \\ \dfrac{r_{ij}(t)}{\sigma(t)}\ , h_{\min}\leq\dfrac{r_{ij}(t)}{\sigma(t)}\leq h_{\max}\end{cases} \tag{18}$$

$h_{\min}=g_0+g(t)$, the value for $g_0$ is set to 1.1, $g(t)=0.1\times\sin\left(\frac{\pi}{2}\times\frac{t}{T}\right)$. $r_{ij}(t)$ is the distance between the best neighboring node and the source node. For example, let us consider node 2 as the best neighbor and the source node as n1. The updated position of the best neighboring node n2 can be written as $x_{jD}=x_{2D}=(x_{21},y_{22},z_{23},w_{24})$ and the initial position of the source node, n1, can be written as $x_{iD}=x_{1D}=(x_{11},y_{12},z_{13},w_{14})$. Therefore, the distance between node $j$ and $i$ can be found out from the equation given below.

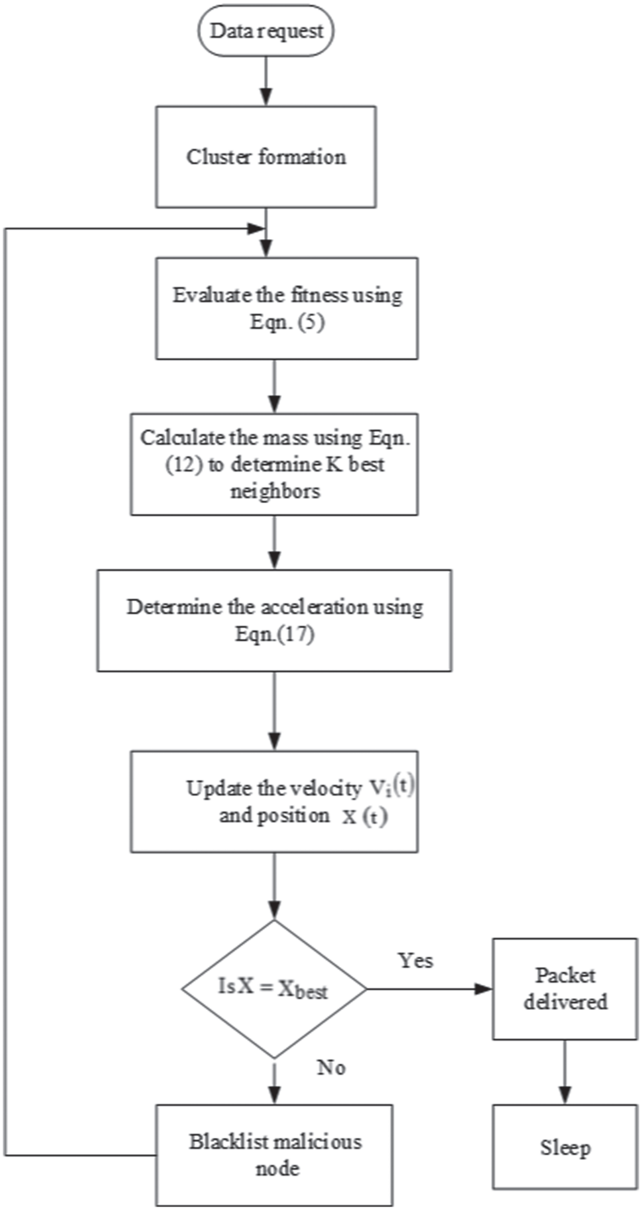
$$r_{ij}(t)=\left\|x_j-x_i\right\|=\sqrt{(x_{21}-x_{11})^2+(y_{22}-y_{12})^2+(z_{23}-z_{13})^{(2)}+(w_{24}-w_{14})^2} \tag{19}$$

On substituting Equation (15) and (19) in Equation 18, $h_{ij}(t)$ can be found out and can be used in finding the acceleration given in Equation (17). The velocity is now updated using $V_i(t)$ = *rand* $v(t)+a(t)$. The updated velocity is now compared with the position of the best node to enhance better trust using the equation, $x(t)=x_i(t)+V_i(t)$. All the calculations and updates are performed continuously until the most secure route is satisfied or when the maximum number of iteration is reached. The flowchart for the proposed method is represented in the below Figure 5.

### *SASR algorithm*

```
Step 1: Initialize the number and velocity of the nodes.
Step 2: Evaluate the fitness of each position $f(x)=\frac{1}{4}[T+E+D+d(i,j)]$ .
Step 3: Rearrange the mass values for each position of the nodes.
Step 4: Find the $K$ best neighboring nodes using $K(t)=N-(N-2)\times\sqrt{\frac{t}{T}}$.
Step 5: Update the position for the best neighboring nodes.
    •Find the acceleration of the nodes, $a(t)=\frac{1}{m_i(t)}[F_i(t)+G_i(t)]$.
    •Where, mass $m_{i(t)}=\frac{M_i(t)}{\sum_{j=1}^{N}M_j(t)}$
    •$F_i(t)=-\alpha\left(1-\frac{t-1}{T}\right)^3 e^{-\frac{20t}{T}\sum_{j\in Kbest} rand_j\left[2\times\left(h_{ij}(t)\right)^{13}-\left(h_{ij}\right)^7\right]\frac{\left(x_j(t)-x_i(t)\right)}{\left\|x_i(t),x_j(t)\right\|_2}}$
    •$G_i(t)=\beta e^{-\frac{20t}{T}}\left[x_{best}(t)-x_{i(t)}\right]$
Step 6: Update the velocity, $V_i(t)$ = rand $v(t)+a(t)$ .
Step 7: Stop when the maximum number of iteration has reached.
```

**FIGURE 5** Flowchart for the Knowledge and Intrusion Detection based Secure Atom Search Routing (KID-SASR) algorithm


Data request
Cluster formation
Evaluate the fitness using Eqn. (5)
Calculate the mass using Eqn. (12) to determine K best neighbors
Determine the acceleration using Eqn.(17)
Update the velocity $V_i(t)$ and position X (t)
Is $X = X_{best}$
Yes
No
Packet delivered
Sleep
Blacklist malicious node

## 5 | PERFORMANCE METRICS

The performance is tested under throughput, delay, packet delivery ratio (PDR), energy, and communication cost and is compared with existing methods.

### 5.1 | Throughput

Throughput is an important factor in analyzing the performance and quality of wireless sensor networks. It is the amount of successful data transmitted in the given time, which is usually expressed in bits per second (bps).

$$T = \frac{P_r}{t}. \tag{20}$$

### 5.2 | Packet Delivery Ratio (PDR)

PDR is the ratio of the number of received packets or messages to the number of sent packets.

$$\mathrm{PDR} = \frac{P_r}{P_s}. \tag{21}$$

### 5.3 | Communication cost

Cost is determined based on the number of nodes that have to calculate trust $C_e(t)$ and packets to be delivered $C_{d(t)}$ in the network lifetime (LT).

$$C = \frac{\sum_{t=0}^{LT} C_e(t) + C_{d(t)}}{LT}. \tag{22}$$

## 6 | EXPERIMENTAL RESULT AND ANALYSIS

The simulation is carried out under the MATLAB simulation platform to determine the effectiveness of the proposed methodology.

### 6.1 | Simulation setup

The simulation is set up in an area of 100m × 100 m with 100 nodes randomly deployed in the network with velocity in the range of 0 to 30m/s. Each node have the initial energy of nodes of 1J, transmission energy of 0.01 J, and the receiving energy of 0.01.J.The BS is located at (50, 50). The antenna used here is Omnidirectional (CR-OMN2409) with a frequency bandwidth of 2.4 GHz. For the vampire attack to happen, we flood the node with RREQ packets and the parameter settings are given in Table 2.

### 6.2 | Routing model

Figure 6 shows the path taken by the nodes under honest condition. Here node 0 is taken as the source node and node 68 is considered as the destination node. The resulting path (0-56-8-80-38-68) taken is observed to be short and optimal due to the absence of the carousal and stretch attack.

Figure 7A shows the path taken for routing without the KID-SASR protocol. Here the nodes 14, 36, 43, 53, 55, and 87 are flooded with RREQ packets to become malicious. Due to this condition, instead of taking the honest path (0-56-8-80-38-64-68), the malicious nodes are misleading the data packet to other nodes to repeat the same route again and again forming a loop (represented in red arrow). This results in more energy consumption, which decreases the lifetime. Figure 7B shows the alternate honest path taken by the KID-SASR algorithm (black arrow). The SASR algorithm finds the best route by estimating the fitness for all the neighboring nodes. Based on the fitness evaluation, the neighboring node having minimum distance, delay, and with maximum trust and energy is selected for data transmission. The path with the worst fitness is eliminated, thus protecting the network from carousal attack.

In Figure 8A, the nodes 14, 36, 43, 53, 55, and 87 are considered as malicious nodes where the nodes 14, 38, and 87 are misleading the packet to take a longer route in an aim to increase the energy consumption. Since the senor node fails to acknowledge the malicious route from the honest path, the path is stretched to its extreme extent. Whereas, Figure 8B shows the shortest optimal path (black arrow) taken by using KID-SASR protocol. This protocol provides the shortest path by avoiding the malicious nodes based on the optimization algorithm provided in Section.3.2

**TABLE 2** Input parameter setting

| Parameters | Value |
|---|---|
| Simulation area | 100m × 100m |
| Simulation time | 200s |
| Base-station position | (50,50) |
| number of nodes | 100 |
| Mobility | Random way |
| Node speed | 0-30m/s |
| Antenna type | Omnidirectional (CR-OMN2409) |
| $E_{elec}$ | 50 nJ/bit |
| $\varepsilon_{amp}$ | 0.0013 pJ/bit/m$^2$ |
| The initial energy of nodes | 1 J |
| Transmission range of sensor | 30m |
| Transmission energy | 0.01 J |
| Receiving energy | 0.01 J |
| The optimum number of clusters | 5 |
| The threshold of CH | 0.05 |
| Trust threshold | 0-0.9 |
| Packet rates | 1 packet per sec |
| Data rate (kbps) | 250 |
| Data packet size | 1500 bytes |
| Control packet size | 25 bytes |
| Traffic type | Constant bit rate |
| Number of malicious nodes | 6 |
| Malicious packet type | RREQ |
| Number of malicious packets sent | 0-10 000 |

Abbreviation: CH: cluster head.

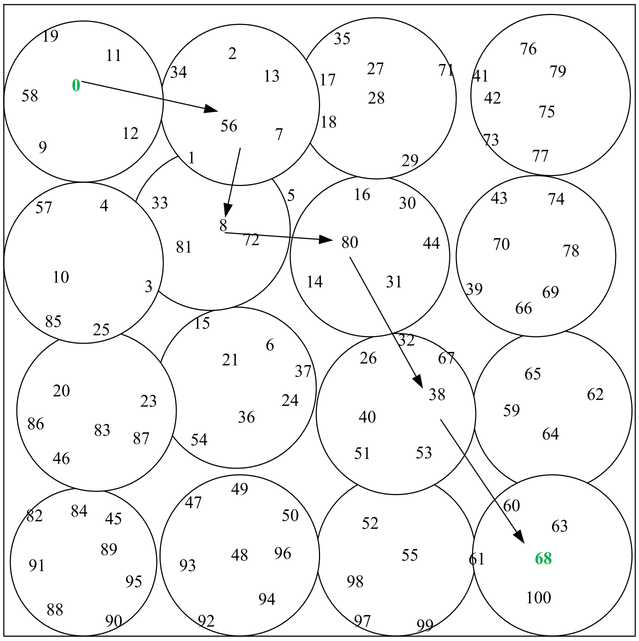


**FIGURE 6** Honest scenario

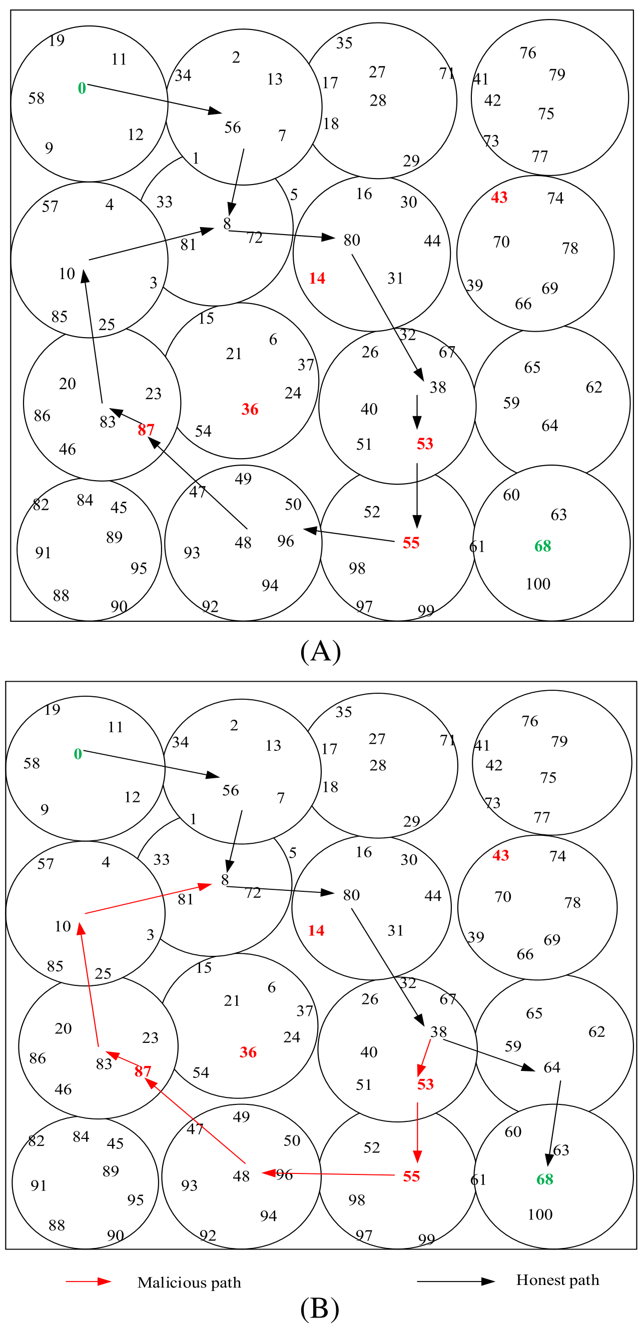


**FIGURE 7** Carousal attack (A) without Knowledge and Intrusion Detection based Secure Atom Search Routing (KID-SASR) and (B) with KID-SASR

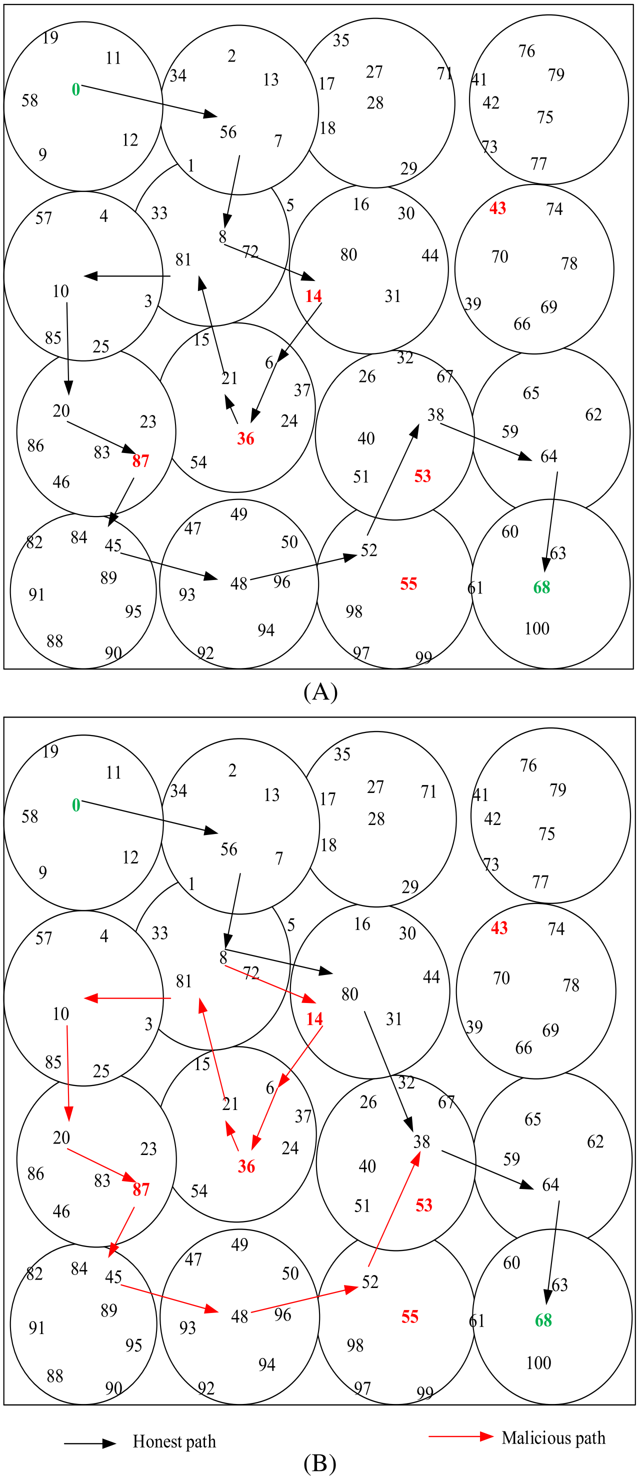
## 6.3 | Comparative analysis

The proposed KID-SASR algorithm is compared with existing EDTM, AF-TNS, TERP, RaRTrust, LEECH, HEED, STRM, EATSRA, Epidemic, Encounter-based, PRoPHET, and PROVEST algorithm for the metrics: communication cost, throughput, trust, energy, PDR, and network lifetime.

### 6.3.1 | Energy consumption analysis

In Figure 9, the graph shows the fraction of energy consumed when transmitting 1, 10, 100, 1000, and 10 000 packets for a various number of hops. It can be observed that the fraction of energy consumed when transmitting 1 packet during carousal attack is very low and when the packet size is increased to 10 000, the energy consumed between 8 hops

**FIGURE 8** Stretch attack (A) without Knowledge and Intrusion Detection based Secure Atom Search Routing (KID-SASR) and (B) with KID-SASR



to16 hops is maximum reaching fraction of 0.045. The network is tested under carousal attack without employing the SASR algorithm. Under this condition, the malicious nodes make the packets to transmit in a closed loop. This leads to more energy consumption.

In Figure 10, the graph shows the average energy consumed for transmitting 1, 10, 100, 1000, and 10 000 packets for a various number of hops under stretch attack. It is observed that the fraction of energy consumed when transmitting 1 packet is very low. When the packet size is increased to 1000, the fraction of energy consumed for 16 hops reaches 0.03 and when 10 000 packets are transmitted, the energy consumed at 16 hops is maximum, reaching 0.045. Under stretch attack, it is observed that the amount of energy consumed is increased due to the abnormal selection of the long route.

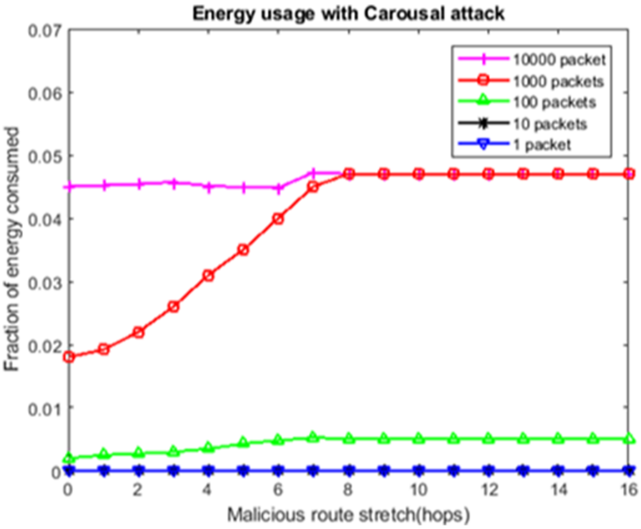


**FIGURE 9** Energy consumed under carousal attack

### 6.3.2 | Delay analysis

The graphical representation for the average delay is given in Figure 11. Here, the proposed KID- SASR protocol is compared with the existing protocols EDTM,[21] TERP,[22] RaRTrust,[23] and AF-TNS[19] under 2 to 10 malicious nodes. From the analysis, EDTM has the highest delay of 24ms for 10 malicious nodes and our proposed KID-SASR algorithm has the lowest delay of 4ms for 10 malicious nodes. This is because the SASR algorithm takes delay into consideration for evaluating the fitness function. As per the condition employed for fitness, the neighboring node, which has the least or minimum delay, makes the best fitness value, whereas the nodes with high delay are considered unfit for packet transmission. This makes the SASR algorithm to achieve packet transmission with less delay.

### 6.3.3 | Packet delivery ratio

In Figure 12, the proposed KID-SASR algorithm is compared with LEECH,[27] HEED,[24] STRM,[25] and EATSRA[20] by sending 100, 200, 300, 400, and 500 packets. From the analysis, it can be observed that the number of packets received is high in the proposed KID-SASR protocol compared with the existing LEACH, HEED, STRM, and EATSRA protocols.

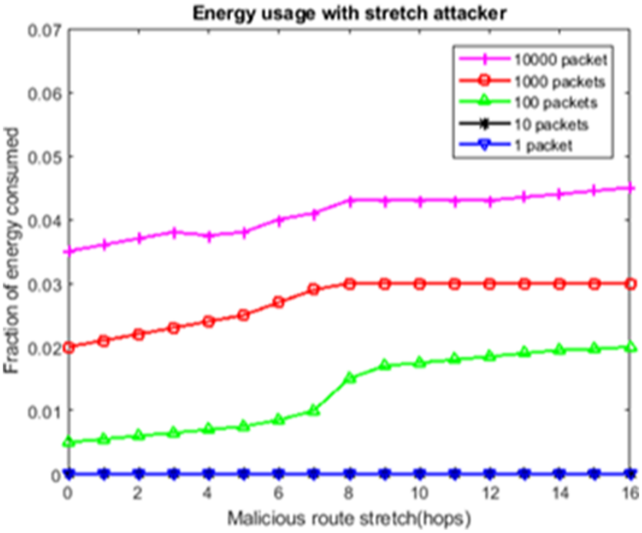
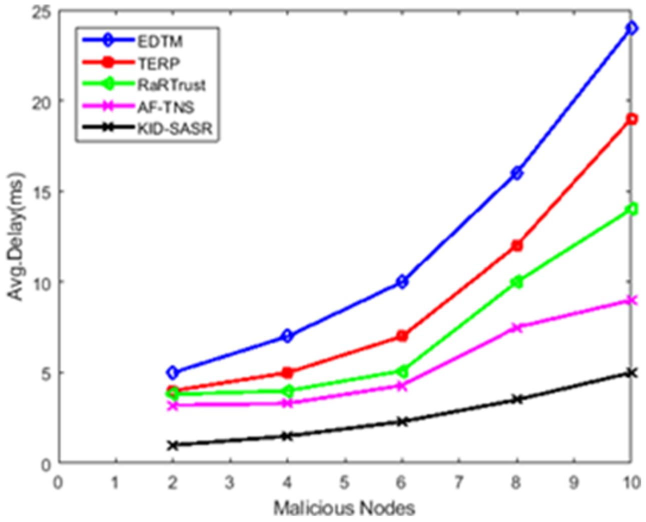


**FIGURE 10** Energy consumed under stretch attack

FIGURE 11 Average delay analysis



The reason for increased the PDR is due to the selection of the routing path based on the direct and indirect trust calculation provided in Equations (2) and (3), respectively. The trust taken into fitness evaluation ensures to select a sensor node that has the high PDR in their previous rounds. This helps to improve the PDR of the KID-SASR algorithm compared with the existing techniques.

### 6.3.4 | Throughput

Figure 13 represents the throughput analysis under a various number of malicious nodes. In the presence of two malicious nodes, the maximum throughput of 200kbps is achieved for the KID-SASR protocol, which is higher than all other existing techniques like EDTM, TERP, RaRTrust, and AF-TNS. The existing EDMT protocol has the least throughput of 120 kbps under two malicious nodes. Since the KB performs the complex computation with faster execution time, the packet is delivered at a maximum throughput rate.

### 6.3.5 | Malicious detection rate

Figure 14 shows the malicious detection rate for the proposed KID-SASR protocol by comparing it with the existing EDTM, TERP, RaRTrust, and AF-TNS protocols for trust update interval of 0.02, 0.04, 0.06, 0.08, and 0.1. When trust is

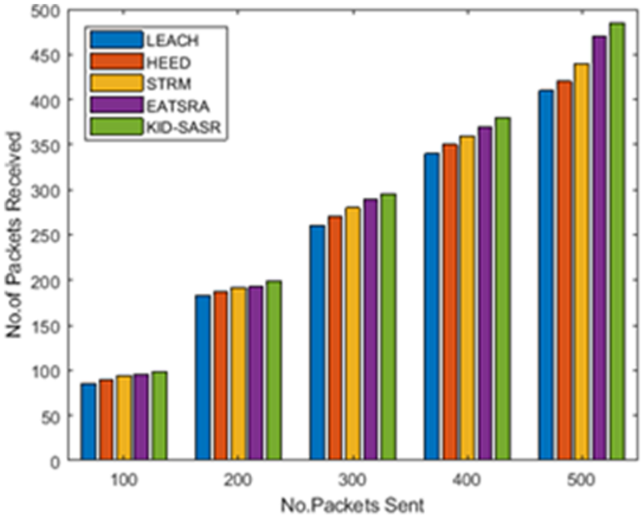


FIGURE 12 Packet delivery analysis

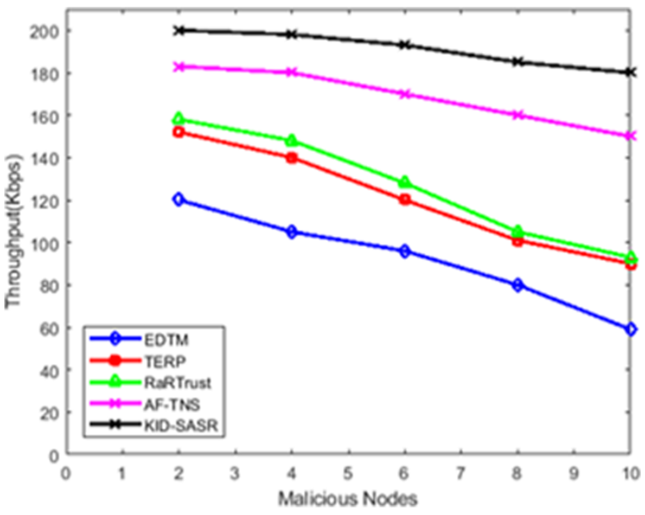


**FIGURE 13** Throughput analysis

updated at the time interval of 0.02s, the detection rate is at the maximum of 50%. When the time interval is increased, the detection rate is gradually decreased. From the analysis, the proposed protocol shows the highest detection rate compared with the existing techniques. The SASR algorithm optimally selects the most secured path by considering trust, delay, energy, and distance between the nodes. Initially, the fitness function is calculated, and then, the position and velocity are updated for the nodes with five best neighboring nodes. Among these five best neighboring nodes, the most optimal solution is taken for packet transmission. This improves the detection of malicious activity in the network.

### 6.3.6 | Network lifetime

The lifetime of a sensor is the operational period of a sensor node until it runs out of energy. Figure 15 represents the network lifetime in the presence of a various number of malicious nodes. For a total lifetime of 600s, the KID-SASR protocol has a maximum lifetime of 480 seconds, whereas the existing EDMT has the least lifetime of 400s under two malicious nodes. As the number of malicious nodes increases, the lifetime of the network is decreased. Also, it is evident that when there are two malicious nodes, the network lifetime of the proposed KID-SASR is 480 seconds and when the number of malicious nodes is increased to 10, the lifetime of the network is decreased to 410 seconds. The existing algorithms fail to detect the malicious activity in the network caused by carousal and stretch attack. These attacks consume more energy from the sensors by transmitting the packets in closed-loop and by a employing longer route to reach the destination. This causes the network lifetime to decrease more rapidly. In contrast, the optimal solution of the SASR

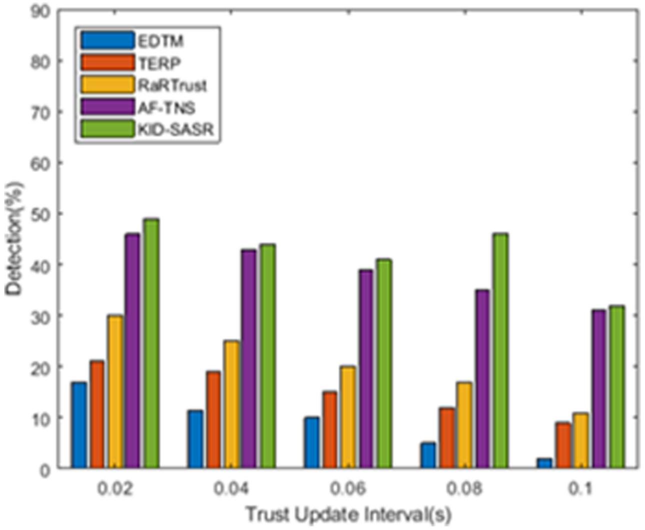


**FIGURE 14** Malicious detection rate

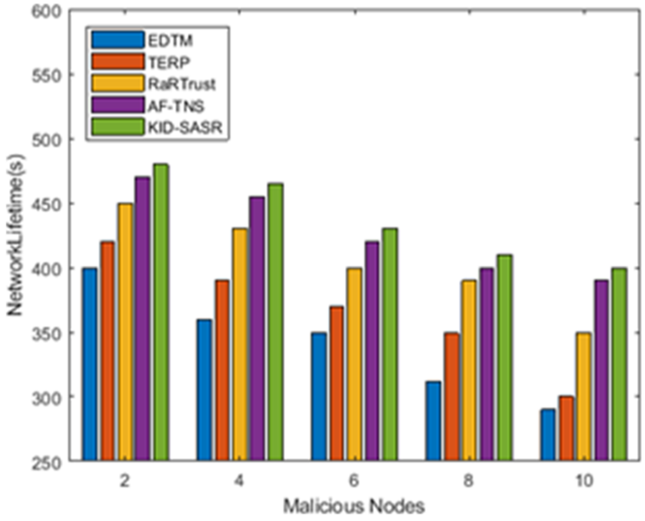


**FIGURE 15** Network lifetime

algorithm helps to select a trusted route by mitigating carousal and stretch attack based on the fitness calculation. This improves the overall lifetime of the network.

### 6.3.7 | Communication cost analysis

Communication cost is the processing time of a node that takes to evaluate the trust degree during each round, which is calculated using Equation (22). Figure 16 shows the communication cost analysis for KID-SASR protocol and the existing Epidemic,[28] Encounter-based,[29] PRoPHET,[30] and PROVEST-HYBRID[17] protocols. From the analysis, the overall communication cost for the KID-SASR protocol is observed to be less than the existing protocols for different trust thresholds.

## 7 | CONCLUSION

The wireless ad hoc sensors are sensitive to malicious attacks especially in sensitive areas like the military and defenses. Hence, in this study, a new routing algorithm called KID-SASR is introduced for security against the vampire attack

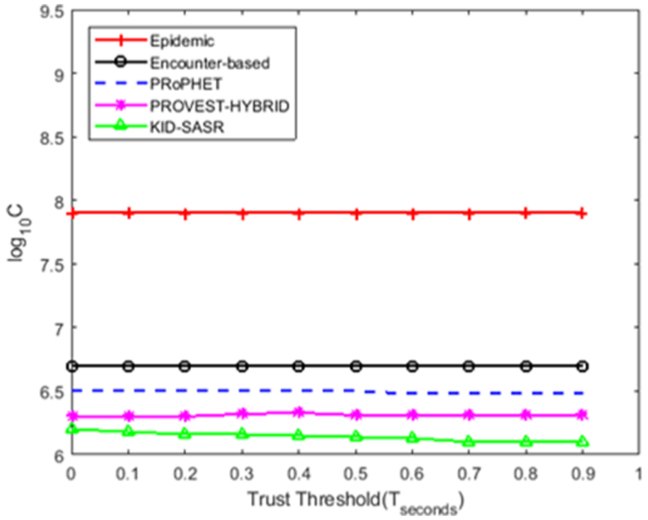


**FIGURE 16** communication cost analysis

and providing the shortest path for packet transmission. From the simulation result, it can be observed that the KID-SASR protocol ensures minimum delay, minimum energy consumption, high throughput, high network lifetime, and low communication cost. Thus, from the result, it is evident that the proposed system is highly efficient and more reliable than the existing protocols. In future works, the mitigation techniques and analysis can be conducted in depth by employing other kinds of attacks such as sleep deprivation attack, directional antenna attack that causes service denial. Moreover, research on highly efficient optimization algorithms can be made to identify and mitigate attacks in topology discovery and packet forwarding phase.

## ORCID

*R. Isaac Sajan* https://orcid.org/0000-0003-3414-5147

## REFERENCES

1. Ye W, Heidemann J, Estrin D. An energy-efficient MAC protocol for wireless sensor networks. In: *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*. Vol.3 n/a: IEEE; 2002:1567-1576.
2. Shaikh FK, Zeadally S. Energy harvesting in wireless sensor networks: a comprehensive review. *Renewable and Sustainable Energy Reviews*. 2016;55:1041-1054.
3. Yetgin H, Cheung KTK, El-Hajjar M, Hanzo LH. A survey of network lifetime maximization techniques in wireless sensor networks. *IEEE Communications Surveys & Tutorials*. 2017;19(2):828-854.
4. Osanaiye O, Choo KKR, Dlodlo M. Distributed denial of service (DDoS) resilience in cloud: review and conceptual cloud DDoS mitigation framework. *Journal of Network and Computer Applications*. 2016;67:147-165.
5. Vasserman EY, Hopper N. Vampire attacks: draining life from wireless ad hoc sensor networks. *IEEE transactions on mobile computing*. 2013;12(2):318-332.
6. Gill K, Yang SH. A scheme for preventing denial of service attacks on wireless sensor networks. In *2009 35th Annual Conference of IEEE Industrial Electronics*. Porto, Portugal: IEEE; 2009:2603-2609.
7. Gunasekaran M, Periakaruppan S. A hybrid protection approaches for denial of service (DoS) attacks in wireless sensor networks. *International Journal of Electronics*. 2017;104(6):993-1007.
8. Osanaiye OA, Alfa AS, Hancke GP. Denial of service defence for resource availability in wireless sensor networks. *IEEE Access*. 2018;6: 6975-7004.
9. Mahfoudh S, Minet P. An energy efficient routing based on OLSR in wireless ad hoc and sensor networks. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops)*. Okinawa, Japan: IEEE; 2008:1253-1259.
10. Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *In ACM SIGCOMM computer communication review, ACM*. 1994;24(4):234-244.
11. Razaque A, Abdulgader M, Joshi C, Amsaad F, Chauhan M. P-LEACH: Energy efficient routing protocol for wireless sensor networks. In: *2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. Farmingdale, NY, USA: IEEE; 2016:1-5.
12. Umakanth B, Damodhar J. Detection of energy draining attack using EWMA in wireless ad hoc sensor networks. *International Journal of Engineering Trends and Technology (IJETT)*. 2013;4(8):3691-3695.
13. Kosunalp S. A new energy prediction algorithm for energy-harvesting wireless sensor networks with Q-Learning. *IEEE Access*. 2016;4: 5755-5763.
14. Sirdeshpande N, Udupi V. Fractional lion optimization for cluster head-based routing protocol in wireless sensor network. *Journal of the Franklin Institute*. 2017;354(11):4457-4480.
15. Rajakumar BR. Lion algorithm for standard and large scale bilinear system identification: a global optimization based on Lion's social behavior. In: *2014 IEEE congress on evolutionary computation (CEC)*. Beijing, China: IEEE; 2014:2116-2123.
16. Mirjalili S, Lewis A. The whale optimization algorithm. *Advances in engineering software*. 2016;95:51-67.
17. Cho JH, Chen R. PROVEST: provenance-based trust model for delay tolerant networks. *IEEE Transactions on Dependable and Secure Computing*. 2018;15(1):151-165.
18. Rajeshkumar G, Valluvan KR. An energy aware trust based intrusion detection system with adaptive acknowledgement for wireless sensor network. *Wireless Personal Communications*. 2017;94(4):1993-2007.
19. AlFarraj O, AlZubi A, Tolba A. Trust-based neighbor selection using activation function for secure routing in wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*. 2018;1-11.
20. Selvi M, Thangaramya K, Ganapathy S, Kulothungan K, Nehemiah HK, Kannan A. An energy aware trust based secure routing algorithm for effective communication in wireless sensor networks. *Wireless Personal Communications*. 2019;1-16.
21. Jiang J, Han G, Wang F, Shu L, Guizani M. An efficient distributed trust model for wireless sensor networks. *IEEE transactions on parallel and distributed systems*. 2015;26(5):1228-1237.
22. Ahmed A, Bakar KA, Channa MI, Haseeb K, Khan AW. TERP: A trust and energy aware routing protocol for wireless sensor network. *IEEE Sensors Journal*. 2015;15(12):6962-6972.
23. Labraoui N, Gueroui M, Sekhri L. A risk-aware reputation-based trust management in wireless sensor networks. *Wireless Personal Communications*. 2016;87(3):1037-1055.

24. Younis O, Fahmy S. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*. 2004;(4):366-379.
25. Das A, Islam MM. SecuredTrust: a dynamic trust computation model for secured communication in multiagent systems. *IEEE Transactions on Dependable and Secure Computing*. 2012;9(2):261-274.
26. Zhao W, Wang L, Zhang Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowledge-Based Systems*. 2019;163:283-304.
27. Heinzelman WB, Chandrakasan AP, Balakrishnan H. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*. 2002;1(4):660-670.
28. Li Y, Hui P, Jin D, Su L, Zeng L. Evaluating the impact of social selfishness on the epidemic routing in delay tolerant networks. *IEEE Communications Letters*. 2010;14(11):1026-1028.
29. Chen R, Bao F, Chang M, Cho JH. Trust management for encounter-based routing in delay tolerant networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. Miami, FL, USA: IEEE; 2010:1-6.
30. Lindgren A, Doria A, Schelén O. *Probabilistic routing in intermittently connected networks*. MobiHoc: In ACM International Symposium on Mobilde Ad Hoc Networking and Computing; 2003.

**How to cite this article:** Isaac Sajan R, Jasper J. Trust-based secure routing and the prevention of vampire attack in wireless ad hoc sensor network. *Int J Commun Syst*. 2020;33:e4341. https://doi.org/10.1002/dac.4341