ORIGINAL RESEARCH

Trust-based secure directed diffusion routing protocol in WSN

Xinying Yu[1] · Fengyin Li[1] · Tao Li[1] · Nan Wu[2] · Hua Wang[1] · Huiyu Zhou[3]

Abstract
Deployed in the monitoring environment, Wireless Sensor Network (WSN) is a measurement and control network composed of miniature and low-cost sensors with sensing, computing, and communication capabilities. The design of the Directed Dif-fusion (DD) routing protocol is one of the key problems in WSN. In order to obtain the confidentiality of sensing data and solve the unreliability of relay nodes in the existing DD routing protocols, this paper designs an Energy Trust Model (ETM) by introducing the remaining energy and trust of a node. We further propose a Trust-based Secure Directed Diffusion Rout-ing protocol (TSDDR) based on the model. The proposed protocol achieves the establishment of a credible communication path and the transmission of confidential data in WSN. Meanwhile, the balance of energy consumption and the privacy of sensing data can be ensured. The performance analysis results show that the TSDDR protocol can effectively defeat against MITM attacks and prevent the malicious nodes' impersonation. At the same time, the protocol achieves secure end-to-end anonymous communication with acceptable energy overhead and computational complexity.

Keywords Wireless sensor network · Directed diffusion routing protocol · Relay node · Energy trust model · Anonymous communication

1 Introduction

Since the concept of Internet of Things (IoTs) was formally proposed in 2005, the idea of the interconnection of everything has gradually penetrated people's lives. From smart home, smart transportation and smart city to agricultural production, medical care, national defense and military, the application field of the Internet of Things has been involved in all aspects. In recent years, the Trust models and e-voting schemes (Li et al. 2019) are often used to assess the trustworthiness of entities in IoTs. Meanwhile, the high-resolution remote sensing image registration (Liu et al. 2019) and the crowd sensing in sensing applications (Jiang et al. 2020) have developed rapidly, which has put forward higher requirements for the Internet of Things.

The wireless sensor network has become an important part of the IoTs, and the sensor nodes are the key part of the wireless sensor network. However, the resource-constrained sensor nodes are usually distributed in various environments in a multi-hop, self-organizing manner and are vulnerable to attacks from internal or external adversaries. Furthermore, the balance of energy consumption, the reliability of routing selection, and the security of data transmission in wireless sensor networks are particularly important.

1.1 Related work

Researchers have designed routing protocols suitable for different practical scenarios. According to the network

✉ Fengyin Li
lfyin318@126.com

Xinying Yu
xyyuqfnu@163.com

Tao Li
litao_ldu@163.com

Nan Wu
qsdjwc@163.com

Hua Wang
qsdjwc@163.com

Huiyu Zhou
hz143@leicester.ac.uk

[1] School of Information Science and Engineering, Qufu Normal University, Rizhao 276826, China

[2] Science and Technology Department, Qufu Normal University, Qufu 273165, China

[3] School of Informatics, University of Leicester, Leicester LE1 7RH, UK

topology, the routing protocols can be divided into two categories, that is flat routing protocols and hierarchical routing protocols (Roy and Das 2014). In flat routing protocols, all sensor nodes are peer-to-peer. The typical flat routing protocols include Directed Diffusion routing protocol (DD) (Zheng et al. 2013), SPIN routing (Feng et al. 2014), and so on. Besides, the main problem to be solved is how to find an energy-saving and low delay route between the source node and the sink node. In hierarchical routing protocols, the source node transfers the sensing data to the cluster head node of the cluster, and a cluster head node can be selected among multiple cluster head nodes, thus forming a layered network structure. The hierarchical routing protocol is mainly concerned with the selection of the cluster head nodes and the formulation of update strategies, so as to achieve the purpose of saving energy. Therefore, the hierarchical routing protocol is suitable for situations requiring centralized data collection.

The DD protocol in flat routing is a typical data-based, query-driven routing mechanism. In the DD protocol, each node only needs to save the information of the neighbor nodes and does not need to maintain the information of the whole network. In addition, the data is sent based on the user's needs, rather than being sent upward as soon as the nodes in the detection area perceive the data. Therefore, the DD routing protocol has the potential advantage of low power consumption and has a high research value in the field of wireless sensor networks. However, the energy consumption of sensor nodes in the original DD routing protocol is unbalanced, and the sensing data is transmitted in plaintext in the network, with poor confidentiality. Therefore, in order to improve network performance, there have been continuous improvements to the DD routing protocol.

Ren et al. proposed a gradient-based limited diffusion algorithm, which performs diffusion in the optimal set of forwarding nodes (Ren et al. 2006). A directed diffusion protocol based on the random key pre-distribution model (Fei et al. 2007) has been proposed in 2007, which can provide point-to-point secure data communication. Dai et al. applied the percolation algorithm to the interest diffusion stage of the directed diffusion protocol to reduce the network overhead (Dai et al. 2010). A cross-layer congestion control method based on directed diffusion routing protocol using the idea of cross-layer design is proposed (Ye et al. 2012), which can effectively relieve congestion and reduce energy consumption. Sengupta et al. proposed a Secure Directed Diffusion (SDD) protocol (Sengupta et al. 2018), which effectively prevents eavesdropping attacks, Sinkhole attacks and Sybil attacks through authentication between adjacent nodes. However, the relay nodes can learn the plaintext data by decryption, which cannot guarantee the confidentiality of the data. Therefore, they further designed an Improved Secure Directed Diffusion (ISDD) protocol (Sengupta et al. 2019), which realizes secure end-to-end data transmission and anonymous communication between nodes. However, the sink node will refuse to provide service due to a large amount of data if most relay nodes on the path are maliciously controlled to send false data to the sink node, which affects the reception of the legitimate data. In other words, the ISDD protocol cannot prevent relay nodes from launching DoS attacks.

1.2 Contributions

The credibility and reliability of relay nodes are very important in the DD protocol. However, if the node with a high trust value is directly selected to build the path, the node with high trust value will cause sharp energy attenuation or even become a "dead node" due to too many communication opportunities, thus affecting the performance of the entire network. Considering the energy limitation of sensor nodes, this paper takes the remaining energy value of nodes into the category of trust value and builds an Energy Trust Model. The energy trust value reflects the trust degree of the nodes in WSNs and provides a guarantee for the selection of reliable nodes in the DD routing protocols. The main contributions of this paper are as follows:

First of all, this paper proposes a Secure Directed Diffusion Routing protocol based on the Energy Trust Model. The Energy Trust Model obtains the energy trust value of a node by weighted summing the direct trust value and the remaining energy to measure the credibility of the node. The proposed protocol establishes a reliable path and transmits sensing data in WSN by selecting relay nodes with high credibility.

Secondly, this paper proposes a secure key distribution method based on the DH protocol. By introducing the DH protocol, the DH session key negotiation between the sink node and each relay node is completed in the Path Reinforcement Phase of the Secure Directed Diffusion Routing protocol, and the secure distribution of the DH session keys is achieved.

Finally, the proposed protocol in this paper transmits sensing data with an idea similar to onion routing. Specifically, this paper utilizes the DH session keys and the pseudonym mechanism to realize multi-layers encryption of sensing data and anonymous communication of nodes in the Data Propagation Phase of the proposed protocol, which can effectively ensure the security of sensing data and the anonymity of nodes.

The remainder of the paper is organized as follows. Section 2 gives the basics used in this paper. An Energy Trust Model is presented in Sect. 3. Section 4 proposes a Trust-based Secure Directed Diffusion Routing protocol. Before summarizing the whole paper in Sect. 6, we analyze the

performance of the routing protocol in terms of security and simulation results in Sect. 5.

2 Preliminaries

2.1 Identity based cryptography (IBC)

The algorithm of Identity Based Cryptography (Zhao et al. 2012) consists of four parts: system initialization, private key extraction, encryption, and decryption.

2.1.1 System initialization

The Private Key Generation center (PKG) selects an appropriate elliptic curve E, a base point P, and two cyclic groups G_1 and G_2 of prime order q, where G_1 is an additive cyclic group and G_2 is a multiplicative cyclic group generated by g. The PKG determines the bilinear pairing $\hat{e} : G_1 \times G_1 \rightarrow G_2$, and chooses two hash functions H_1 and H_2. $H_1 : \{0,1\}^* \rightarrow G_1$ is used to map the user's Id to G_1 and $H_2 : \{0,1\}^n \rightarrow G_2$, is used to map the elements on G_2 to plaintext space M. The PKG also chooses a random value $m_s \in Z_q^*$ as the system master key and computes the system public key $X = m_s P$. Finally, the PKG publishes the system parameter $\{q, g, P, X, G_1, G_2, \hat{e}, H_1, H_2\}$.

2.1.2 Private key extraction

The PKG generates public key $Q = H_1(Id)$ and private key $PK = m_s Q$ for the user with the identity Id and sends PK to the user.

2.1.3 Encryption

User A selects a random value $r \rightarrow Z_q^*$, computes $C_1 = rP$, $C_2 = m \oplus H_2(\hat{e}(Q_B, X)^r)$, and sends cipher text (C_1, C_2) to user B, where m is the plaintext message and Q_B is the public key of user B.

2.1.4 Decryption

After receiving the cipher text, user B restores the message $m = C_2 \oplus H_2(\hat{e}(PK_B, C_1))$, where the PK_B is the private key of user B.

2.2 Bilinear pairing

Bilinear pairing (Zhang et al. 2004) can be described by $(q, G_1, G_2, \hat{e})$, where G_1 is an additive cyclic group whose order is the prime q, and G_2 is a multiplicative cyclic group with the same order q. Let $\hat{e} : G_1 \times G_1 \rightarrow G_2$ be a map with the following properties:

2.2.1 Bilinearity

For all $P, Q \in G_1$ and $a, b \in Z_q^*$, we have:

$$\hat{e}(aP, bQ) = \hat{e}(P, bQ)^a = \hat{e}(aP, Q)^b = \hat{e}(P, Q)^{ab} \tag{1}$$

2.2.2 Non-degeneracy

If P is a generator of G_1, then the follows holds:

$$\forall P \in G_1 and P \neq 0 \Rightarrow \hat{e}(P, P) = G_2 \tag{2}$$

2.2.3 Computability

There is an efficient algorithm to compute $\hat{e}(P, Q)$for all $P, Q \in G_1$.

2.3 DH key exchange protocol

The effectiveness of Diffie–Hellamn algorithm depends on the difficulty of calculating discrete logarithm problems (Li et al. 2014; Wang et al. 2019).

Firstly, let's define the Discrete Logarithm. Assuming that a is a primitive root of the prime number q, its power can produce all integers between 1 and $q-1$. That is, $a \bmod q, a^2 \bmod q, \ldots, a^{q-1} \bmod q$ are different, which is a permutation between integers 1 and $q-1$. For any integer b and the primitive root a of prime q, we can find the unique exponent $i (0 \leq i \leq q-1)$ such that

$$b = a^i \bmod q \tag{3}$$

The exponent i is called the discrete logarithm of b with a as the base module q and is denoted as $dlog_{a,q} b$.

The Discrete-Logarithm Problem in a cyclic group G with generator g is to compute $\log_g h$ for a uniform element $h \in G$. The Discrete-Logarithm assumption is simply the assumption that there exists a G for which the Discrete-Logarithm Problem is hard. In short, it is easy to compute $h = g^x$ given x, but it is hard to compute x given $h = g^x$.

In this paper, the multiplicative cyclic group G_2 is used as the number field of the DH algorithm. Assuming that users A and B want to negotiate a key. User A selects a random integer $x \in Z_q^*$ and calculates his DH public key $P_A = g^x \bmod q$, and then sends it to B. User B also chooses a random integer $y \in Z_q^*$ and calculates his public key $P_B = g^y \bmod q$, and then sends it to A. Users A and B keep x and y secret respectively, and finally calculate the DH session key:

$$K_A = (P_B)^x \bmod q \tag{4}$$

$$K_B = (P_A)^y \bmod q \tag{5}$$

The two results are the same.

2.4 Directed diffusion (DD) routing protocol

2.4.1 Interest message

The interest message describes the information that users want to query in the form of a set of attribute values and floods the wireless sensor network starting from the sink node. The attribute combination of an interest message should include the detection object, the location of the detection area, the start time of data collection, the transmission signal period, and the signal strength, etc. (Roy and Das 2014). The source node also uses a set of matching attribute values to represent the collected data.

2.4.2 Gradient

The gradient is a data structure used to transmit data. The direction of the gradient is the direction of data transmission, which is opposed to the direction of interest propagation. The gradient value reflects the similarity between the sensing data and the interest message, which is one of the measurement criteria of path selection.

Directed diffusion routing protocol is a query-based routing mechanism, which consists of Interest Propagation Phase, Gradient Establishment Phase, Path Reinforcement Phase, and Data Propagation Phase. In the Interest Propagation Phase, the sink node floods interest messages to all nodes in the target area. In the Gradient Establishment Phase, each node establishes a data transmission gradient with neighbor nodes that send interest messages. In the Path Reinforcement Phase, along the data transmission gradient, the source node floods the probe data to the sink node. After receiving the probe data from multiple paths, the sink node selects an optimal path for subsequent data transmission according to a certain reinforcement mechanism (such as lower delay or shorter hop). In the Data Propagation Phase, the source node sends the data it collected to the sink node along the enhanced path at a high speed.

3 Energy trust model

The Energy Trust Model (ETM) weighted summing the Direct Trust Value and the Energy Specification Value to obtain the Energy Trust Value of a node. In the proposed Energy Trust Model, we utilize the simplified Beta trust model (Ye et al. 2019) to calculate the Direct Trust Value. Our major contribution lies in the Energy Trust Model where we defined a new notion of Energy Specification Value and proposed its computing method using the remaining energy of a node.

Before formally defining the proposed model, we first prepare some initial settings. Supposing that there are n sensor nodes in a certain area when the wireless sensor network is initially formed. Each node saves a neighbor nodes list, which stores the Id, Direct Trust Value, Remaining Energy Value, Energy Specification Value, and Energy Trust Value of neighbor nodes. The initial direct trust value of each node is set to 0.5, and the initial energy value is E_0. The initial state of the neighbor node list is like in Table 1.

3.1 Direct trust value

The Direct Trust Value is between [0, 1], with 0 indicates complete distrust and 1 indicates complete trust. As the number of interactions increases, the Direct Trust Value changes.

Definition 1 Direct Trust Value (*DT*)

First setting a time period t. We assume that in the time period t, the node i actively communicates with node j for a total of $\alpha + \beta$ times, in which the successful interaction is α times and the failure interaction is β times. The Direct Trust Value of i to j is defined as:

$$DT_{i,j}(t) = \frac{\alpha + 1}{\alpha + \beta + 2}\left(1 - \frac{\beta}{W}\right)\left(1 - \frac{1}{\alpha + \delta}\right) \tag{6}$$

The $(1 - \frac{\beta}{W})$ is a penalty function, in which W is the total communication times between node i and j. The $(1 - \frac{1}{\alpha+\delta})$ is a tuning function, in which δ is a positive constant used to adjust the speed close to 1.

3.2 Energy specification value

Definition 2 Energy Consumption Value (*EC*)

Assuming that the node i sends k-bits data to node j and the distance between the two nodes is d_{ij}. So the energy consumption of node i is defined as:

$$E_{cons} = \begin{cases} \left(E_{elec} + d_{ij}^2 E_{amp1}\right)k, & d_{ij} < d_0; \\ \left(E_{elec} + d_{ij}^4 E_{amp2}\right)k, & d_{ij} \geq d_0. \end{cases} \tag{7}$$

The E_{elec} is the energy consumed by each bit of data received by a node and d_0 is the distance threshold. E_{amp1} and E_{amp2}

Table 1 The initial state of the neighbor nodes list

Id	Direct trust value	Remaining energy value	Energy specification value	Energy trust value
Id_i	0.5	E_0	1	$0.5\lambda_1 + \lambda_2$

represent the unit energy consumption of the power amplifier in the Free Space Model and the Multipath Attenuation Model (Ye et al. 2019), respectively. When the communication distance is less than the threshold, the propagation consumption has a quadratic relationship with the distance. While on the opposite, the propagation consumption will have a quartic relationship with the distance. Thus, the greater the communication distance, the more energy is consumed.

Definition 3 Energy Remaining Value (*ER*)

The Energy Remaining Value of node i after sending k-bits data is defined as:

$$E_{now}^{t} = E_{now}^{t-1} - E_{cons} \tag{8}$$

The E_{now}^{t} is the remaining energy of node i in the current time period while the E_{now}^{t-1} is the remaining energy of nodei in the previous period. The initial energy value of each node is E_0, that is, when $t = 0$, $E_{now}^{t} = E_0$. Node i stores E_{now}^{t} locally, updates it in real time according to the forwarded data, and periodically sends its latest remaining energy to its neighbor nodes.

Definition 4 Energy Specification Value (*ES*)

The Energy Specification Value is the ratio of the node's current remaining energy E_{now}^{t} to the initial energy E_0, where $ES \in [0, 1]$ and the closer the *ES* is to 1, the more remaining energy the node has.

The update process of the *ES* is as follows. Similar to the update of the *DT*, the update of the *ES* is also in the unit of time period t. After time t, the *ES* of node i to node j is

$$ES_{i,j}(t) = \frac{E_{now}^{t}}{E_0} = \frac{E_{now}^{t-1} - E_{cons}}{E_0} \tag{9}$$

It can be seen that when the initial energy E_0 is fixed, the size of the *ES* is related to the current remaining energy of the node. The more remaining energy, the larger the energy specification value, and the greater the probability of participating in data transmission. In this way, the energy consumption of the entire network is relatively balanced, and the average life of each node is also extended.

3.3 Energy trust value

Definition 5 Energy Trust Value (*ET*)

The Energy Trust Value comprehensively considers the node's Direct Trust Value *DT* and Energy Specification Value *ES*, so as to obtain the node's energy-based comprehensive trust value. This value reflects the overall reliability and trustworthiness of the node, which is defined as follows:

$$ET_{i,j}(t) = \lambda_1 DT_{i,j}(t) + \lambda_2 ES_{i,j}(t) \tag{10}$$

λ_1 and λ_2 are weight factors and $\lambda_1 + \lambda_2 = 1$. The Energy Trust Value *ET* is updated periodically with *DT* and *E*S, that is, each node periodically updates its own neighbor nodes list.

4 The trust-based secure directed diffusion routing protocol (TSDDR)

The Secure Directed Diffusion Routing Protocol based on the Energy Trust Model is referred to as the Trust-based Secure Directed Diffusion Routing Protocol (TSDDR). The proposed protocol includes Predeployment Phase, Interest Propagation and Gradient Establishment Phase, Path Reinforcement Phase, and Data Propagation Phase.Our revised protocol proposals changes in the Path Reinforcement Phase and Data Propagation Phase. By introducing the ETM into WSNs, the TSDDR protocol can use the Energy Trust Value as a metric to select trusted nodes in the Path Reinforcement Phase. In addition, the DH key exchange protocol is also introduced at this phase to complete the key negotiation between the sink node and the trusted node. The generated key is used to encrypt data during the Data Propagation Phase.

In the schematic diagrams of the following phases, the size of the number represents the sequence of operations, and the same number represents that operations can occur synchronously. The main notations used in the TSDDR protocol and their meanings are shown in Table 2.

4.1 Predeployment phase

This phase occurs between the trusted Private Key Generation center and each sensor node newly added to the

Table 2 Notations

Notation	Meaning
m_s	The system master key selected by PKG
Id_i	The identity of node i
Loc_i	The location of node i
Q_i	The public key of node i
PK_i	The private key of node i
P_i	The DH public key of node i
Z_q^*	Integer multiplication group of order q
RK_i	The random private key of node i
PN_i	The random pseudonym of node i
$SK_{i,j}$	The shared key of node i and node j
$K_{i,j}$	The DH session key of node i and node j

network. The PKG assigns a private key to each node using the Identity Based Cryptography, which is used to calculate the random private key and shared key of the node in the Data Propagation Phase.

We assume that each node (say, node i) is provided with a unique, integer-valued and non-zero identity denoted by Id_i. When the node i joins the network, it sends its own identity Id_i to PKG. Correspondingly, the PKG calculates the public key Q_i and private key PK_i for node i (as shown in Formulae 11 and 12) and sends the system parameters and private key to node i. The specific communication process is shown in Fig. 1.

$$Q_i = H_1(Id_i) \tag{11}$$

$$PK_i = m_s Q_i \tag{12}$$

4.2 Interest propagation and gradient establishment phase

After Predeployment Phase, each node in the network is assigned the system parameters and a private key. The Interest Propagation is started by the sink node(*SN*) flooding an Interest package of the form $< Interest, Id_{SN}, Loc_{SN} >$ containing the attribute-value pairs *Interest*, the identity Id_{SN} of *SN*, and the location Loc_{SN} of *SN*. Generally speaking, when a node j receives an Interest package $< Interest, Id_i, Loc_i >$ from node i, the node j will update its interest list and flood an Interest package $< Interest, Id_j, Loc_j >$ to its neighbor nodes. In the Interest package, the set of the attribute-value pairs is essentially an Interest Message defined in Sect. 2.4, which describes the information that the user wants to query and is represented by *Interest*. The phase continues until all nodes in the network receive an Interest package.

The Gradient Establishment Phase is synchronized with the Interest Propagation Phase, thus both the phases complete together. In detail, at the same time of the interest propagation, the node j that has received an Interest package from node i will build a gradient towards i. And then the node j unicasts a three-tuple package $< Gradient, Id_j, Loc_j >$ to i (as shown in Fig. 2). The gradient defined in Sect. 2.4 is a data structure used to store routing information and transmit data, whose direction is opposite to that of the interest propagation. As this phase continues, the gradient values are also updated in the cache of each node. The completion of the Interest Propagation Phase means that multiple paths established by gradients are formed between the source and the sink node. After that, the source node floods the probe data to the sink node along the gradient direction.

4.3 Path reinforcement phase

In this phase, a trusted path is established from the sink node to the source node (as shown in Fig. 3). In the process of path establishment, the sink node negotiates a DH session key with each new node that joins the path (as shown in Fig. 4). This phase uses the *ET* as the enhancement mechanism to select relay nodes. The higher the *ET* of a node, the more likely it is to be selected as a relay node. When a path is composed of a group of highly reliable nodes, the data propagation is more stable and reliable.

As shown in Fig. 3, the trusted path establishment process is as follows. The establishment process of the trusted path starts from the sink node *SN*. When a node i that has joined the path selects the next relay node from its neighbor node list, the following two checks are performed: (1) whether there is a source node in the neighbor node list of node i: if so, the path to the source node is directly established, otherwise, node i selects the node with the highest *ET* value from its neighbor node list; (2) whether the node is a sink

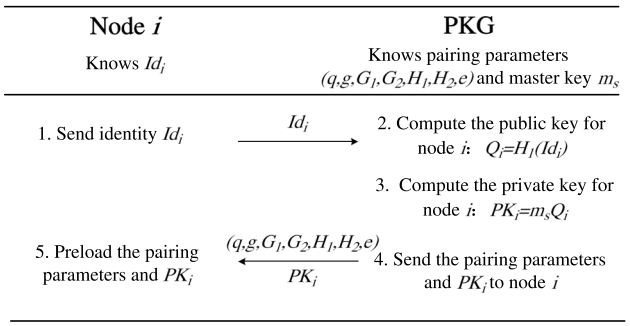

Fig. 1 Predeployment phase

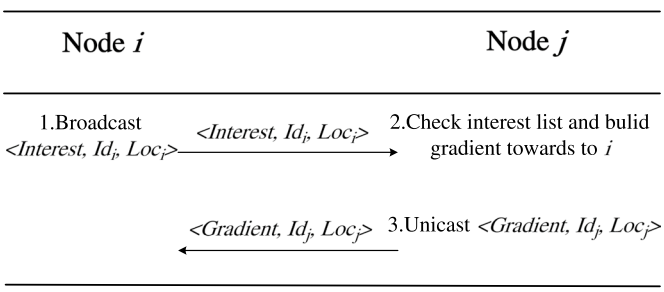

Fig. 2 Interest propagation and gradient establishment phase

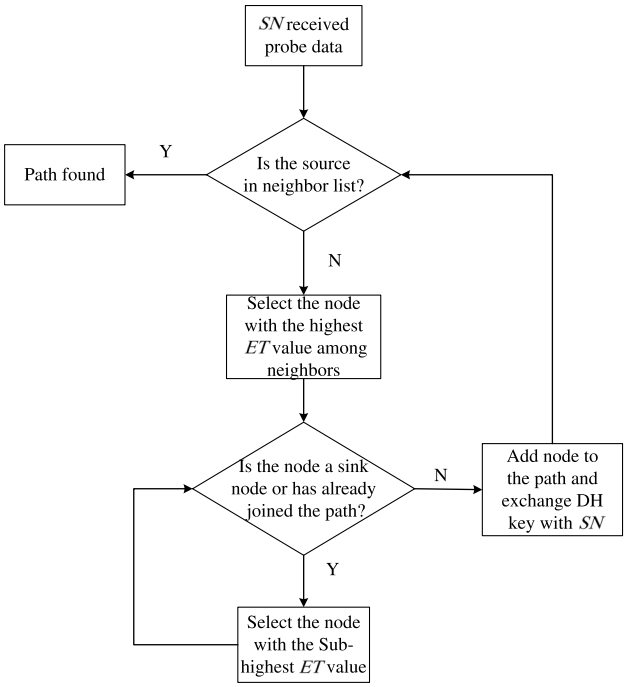

Fig. 3 The path establishment process

node or has joined the path: if so, node i selects the node with the sub-highest ET value and performs the same check, otherwise the node will be added to the path to continue the first check. Until a path from the sink node to the source node is established, suppose the established trusted path is $SN \rightarrow A \rightarrow B \rightarrow S$.

As shown in Fig. 4, the DH key negotiation process is as follows. In the path establishment process, each time a new relay node is added, the DH algorithm is used to calculate the session key with the sink node. The specific process is as follows.

1. SN randomly selects a positive integer $x \in Z_q^*$ and calculates its DH public key P_{SN}:

$$P_{SN} = g^x \bmod q \tag{13}$$

2. SN sends the package $<$ *Reinforcement* $>$ and P_{SN} to node A.
3. After receiving the information, A randomly selects a positive integer $y \in Z_q^*$ and calculates its DH public key P_A:

$$P_A = g^y \bmod q \tag{14}$$

4. A sends the P_A along the path $(A \rightarrow SN)$ to the sink node SN.
5. A and SN calculate their DH session keys $K_{A,SN}$ and $K_{SN,A}$ respectively:

$$K_{A,SN} = (P_{SN})^y \bmod q \tag{15}$$

$$K_{SN,A} = (P_A)^x \bmod q \tag{16}$$

6. A sends the package $<$ *Reinforcement* $>$ and P_{SN} to node B.
7. After receiving the information, node B randomly selects a positive integer $z \in Z_q^*$ and calculates its DH public key P_B:

$$P_B = g^z \bmod q \tag{17}$$

8. B sends the P_B along the path $(B \rightarrow A)$ to the node A.
9. A sends the P_B along the path $(A \rightarrow SN)$ to the sink node SN.
10. B and SN calculate their DH session keys $K_{B,SN}$ and $K_{SN,B}$ respectively:

$$K_{B,SN} = (P_{SN})^z \bmod q \tag{18}$$

$$K_{SN,B} = (P_B)^x \bmod q \tag{19}$$

11. B sends the package $<$ *Reinforcement* $>$ and P_SN to the source node S.
12. After receiving the information, node S randomly selects a positive integer $w \in Z_q^*$ and calculates its DH public key P_S:

$$P_S = g^w \bmod q \tag{20}$$

13. S sends the P_S along the path $(S \rightarrow B)$ to the node B.
14. B sends the P_S along the path $(B \rightarrow A)$ to the node A.
15. A sends the P_S along the path $(A \rightarrow SN)$ to the sink node SN.
16. S and SN calculate their DH session keys $K_{S,SN}$ and $K_{SN,S}$ respectively:

$$K_{S,SN} = (P_{SN})^w \bmod q \tag{21}$$

$$K_{SN,S} = (P_S)^x \bmod q \tag{22}$$

4.4 Data propagation phase

In this phase, the TSDDR protocol will use the DH session keys negotiated between the sink node and relay nodes, and the shared keys generated between the adjacent relay nodes to produce multiple encryption layers thereby providing end to end data security. Specifically, the source node transmits the original data to the sink node along the path after multi-layers encryption, and the sink node gets the plaintext through multi-layers decryption. The whole encryption and decryption operations are similar to the process of the onion routing (El Mougy and Sameh 2018; Hiller et al. 2019).

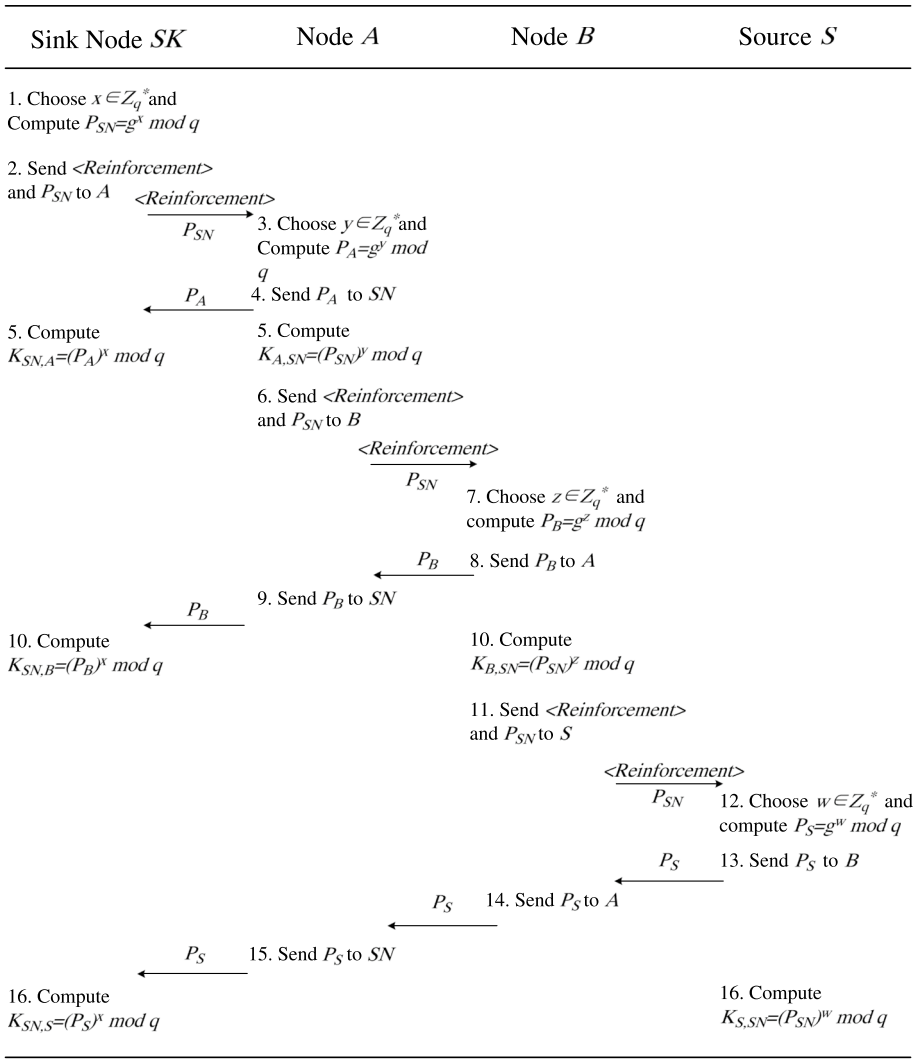

Fig. 4 The path establishment process

In the sequel, each relay node in the established path will calculate a random pseudonym to conceal its real identity thereby achieving anonymous communication. The random pseudonym is obtained by multiplying a random integer with the public key of a node and will be sent to the next relay node along with the encrypted data. The receiving node will retrieve the random pseudonym to calculate the shared key with the sending node. Figure 5 depicts each of the steps of the Data Propagation Phase for an example data path $S \rightarrow B \rightarrow A \rightarrow SN$.

The source node S does the following before transmitting the original data.

1. S calculates the public key Q_B according to the identity Id_B of its direct successor node B:

$$Q_B = H_1(Id_B) \tag{23}$$

2. S chooses a random integer $n_1 \in Z_q^*$, calculates a random pseudonym PN_S and a random private key RK_S:

$$PN_S = n_1 Q_S \tag{24}$$

$$RK_S = n_1 PK_S \tag{25}$$

3. S calculates the shared key $SK_{S,B}$ with B using bilinear pairing:

$$\begin{aligned} SK_{S,B} &= \hat{e}(RK_S, Q_B) = \hat{e}(n_1 PK_S, Q_B) \\ &= \hat{e}(n_1 m_s Q_S, Q_B) = \hat{e}(Q_S, Q_B)^{n_1 m_s} \end{aligned} \tag{26}$$

4. S uses the DH session key $K_{S,SN}$ with the sink node SN to encrypt the plaintext m:

$$c_1 = H_2(K_{S,SN}) \oplus m \tag{27}$$

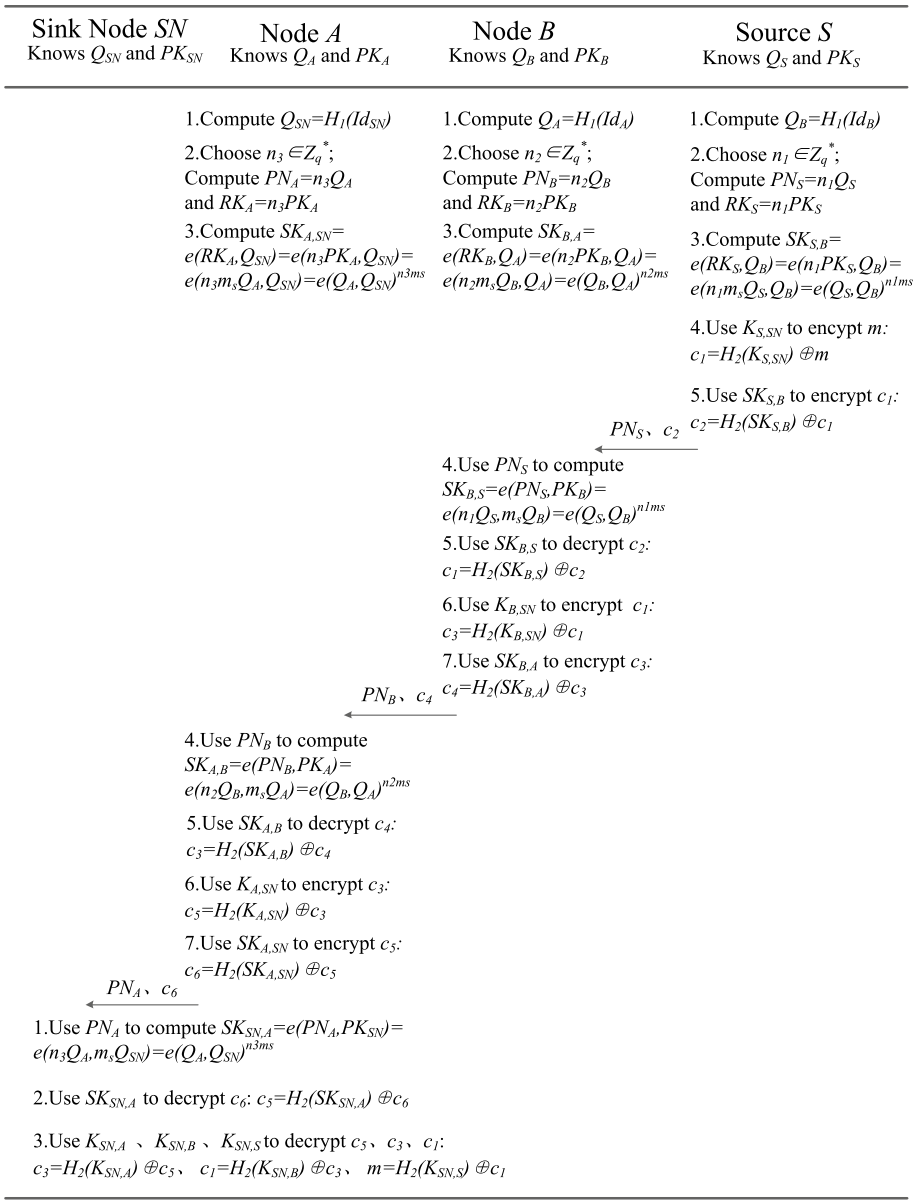

Fig. 5 The data propagation process

5. S encrypts c_1 using the shared key $SK_{S,B}$ with its successor node B:

$$c_2 = H_2(SK_{S,B}) \oplus c_1 \tag{28}$$

Next, S sends its pseudonym PN_S and the cipher text c_2 which has two layers of encryption to B. B performs similar operations to S among steps 1 to 3 before receiving the message.

1. B calculates the public key Q_A according to the identity Id_A of its direct successor node A:

$$Q_A = H_1(Id_A) \tag{29}$$

2. B chooses a random integer $n_2 \in Z_q^*$, calculates a random pseudonym PN_B and a random private key RK_B:

$$PN_B = n_2 Q_B \tag{30}$$

$$RK_B = n_2 PK_B \tag{31}$$

3. B calculates the shared key $SK_{B,A}$ with A using bilinear pairing:

$$\begin{aligned} SK_{B,A} &= \hat{e}(RK_B, Q_A) = \hat{e}(n_2 PK_B, Q_A) \\ &= \hat{e}(n_2 m_s Q_B, Q_A) = \hat{e}(Q_B, Q_A)^{n_2 m_s} \end{aligned} \tag{32}$$

4. After receiving the message from S, B calculates the shared key $SK_{B,S}$ with S according to the pseudonym PN_S:

$$\begin{aligned} SK_{B,S} =&\hat{e}(PN_S, PK_B) = \hat{e}(n_1 Q_S, m_s Q_B) \\ =&\hat{e}(Q_S, Q_B)^{n_1 m_s} \end{aligned} \tag{33}$$

5. B decrypts c_2 with $SK_{B,S}$ to obtain c_1:

$$c_1 = H_2(SK_{B,S}) \oplus c_2 \tag{34}$$

6. B uses the DH session key $K_{B,SN}$ with the sink node SN to encrypt c_1:

$$c_3 = H_2(K_{B,SN}) \oplus c_1 \tag{35}$$

7. B uses the shared key $SK_{B,A}$ with A to encrypt c_3:

$$c_4 = H_2(SK_{B,A}) \oplus c_3 \tag{36}$$

Then, B sends its pseudonym PN_B and cipher text c_4 which has three layers of encryption to A. A performs the similar actions to B among steps 1 to 3 before receiving the message.

1. A calculates the public key Q_{SN} according to the identity Id_{SN} of its direct successor node SN:

$$Q_{SN} = H_1(Id_{SN}) \tag{37}$$

2. A chooses a random integer $n_3 \in Z_q^*$, and calculates a random pseudonym PN_A and a random private key RK_A :

$$PN_A = n_3 Q_A \tag{38}$$

$$RK_A = n_3 PK_A \tag{39}$$

3. A calculates the shared key $SK_{A,SN}$ with SN using bilinear pairing:

$$\begin{aligned} SK_{A,SN} =&\hat{e}(RK_A, Q_{SN}) = \hat{e}(n_3 PK_A, \\ Q_{SN}) =&\hat{e}(n_3 m_s Q_A, Q_{SN}) = \hat{e}(Q_A, Q_{SN})^{n_3 m_s} \end{aligned} \tag{40}$$

4. After receiving the message from B, A calculates the shared key $SK_{A,B}$ with B according to the pseudonym PN_B:

$$\begin{aligned} SK_{A,B} =&\hat{e}(PN_B, PK_A) = \hat{e}(n_2 Q_B, m_s Q_A) \\ =&\hat{e}(Q_B, Q_A)^{n_2 m_s} \end{aligned} \tag{41}$$

5. A uses the shared key $SK_{A,B}$ to decrypt c_4 to get c_3:

$$c_3 = H_2(SK_{A,B}) \oplus c_4 \tag{42}$$

6. A encrypts c_3 with the DH session key $K_{A,SN}$ negotiated with the sink node SN:

$$c_5 = H_2(K_{A,SN}) \oplus c_3 \tag{43}$$

7. A uses the shared key $SK_{A,SN}$ with SN to encrypt c_5:

$$c_6 = H_2(SK_{A,SN}) \oplus c_5 \tag{44}$$

Then, A sends its pseudonym PN_A and cipher text c_6 which has four layers of encryption to the sink node SN.

1. After receiving the message, SN calculates the shared key $SK_{SN,A}$ with A according to the pseudonym PN_A:

$$\begin{aligned} SK_{SN,A} =&\hat{e}(PN_A, PK_{SN}) = \hat{e}(n_3 Q_A, m_s Q_{SN}) \\ =&\hat{e}(Q_A, Q_{SN})^{n_3 m_s} \end{aligned} \tag{45}$$

2. SN uses $SK_{SN,A}$ to decrypt c_6 to get c_5:

$$c_5 = H_2(SK_{SN,A}) \oplus c_6 \tag{46}$$

3. SN decrypts c_5 with $K_{SN,A}$, $K_{SN,B}$ and $K_{SN,S}$ respectively to retrieve the plaintext:

$$c_3 = H_2(K_{SN,A}) \oplus c_5 \tag{47}$$

$$c_1 = H_2(K_{SN,B}) \oplus c_3 \tag{48}$$

$$m = H_2(K_{SN,S}) \oplus c_1 \tag{49}$$

5 Performance analysis

5.1 Security analysis

5.1.1 Anonymous communication

During the data propagation, each relay node on the data path generates a fresh pseudonym to guarantee anonymous communication. We analyze the security goal by an example, a relay node i on the data path generates a random pseudonym PN_i using $PN_i = nQ_i = nH_1(Id_i)$, in which $n \in Z_q^*$ is a random integer, $H_1() : \{0,1\}^* \rightarrow G_1$ is a one-way hash function, and G_1 is a cyclic group of prime order. Therefore, the pseudonym $PN_i \in G_1$ completely blinds the real identity information Id_i of node i. Furthermore, when the node i acts as a relay node for multiple paths, it only knows the pseudonyms of its predecessor nodes but cannot precisely distinguish which one it is. That is, a relay node on the data path only knows the next hop but cannot identify the previous hop node, which can further realizes anonymous communication between the nodes.

5.1.2 End to end data security

As is shown in the Data Propagation Phase, the original plaintext data sent by the source node has multiple layers of encryption while arrived at the sink node. Furthermore,

the innermost encryption layer of the encrypted data is calculated with the DH session key which is only known by the source node and the sink node. Therefore, the original plaintext data can only be decrypted by the sink node using its DH session key. Each relay node on the data path cannot retrieve the original plaintext data, thus our proposed protocol can ensure the end to end data security.

5.1.3 No impersonation

We consider an external adversary Adv with Id_{Adv}, Q_{Adv}, and PK_{Adv} who wants to impersonate a legal relay node say B on the path $S \rightarrow B \rightarrow A \rightarrow SN$. Assume that the adversary has obtained the pseudonym PN_S of source node S and the public key Q_B of node B. The adversary would need to compute the shared key $SK_{B,S} = \hat{e}(PN_S, Q_B)^{m_s}$ after receiving the ciphertext from S to behave as the node B. However, the adversary has no idea of m_s which is only known by the PKG, so he cannot calculate the key $SK_{B,S}$ shared between B and S. It is worthy to note that, it is computationally infeasible for the adversary to deduce m_s through $PK_{Adv} = m_s Q_{Adv}$ given PK_{Adv} and Q_{Adv}, which is exactly the Discrete Logarithm Problem(DLP) defined in Sect. 2.3. Therefore, under the DLP assumption in the additive cyclic group G_1, the impersonation of any other node is infeasible.

5.1.4 Defending against man-in-the-middle (MITM) attack

We will analyze the MITM Attack through an example. Let's consider a malicious node Adv that attempts to initiate a MITM Attack between the sink node SN and any legal node say node A on the data path $S \rightarrow B \rightarrow A \rightarrow SN$. In this scene, assuming that the malicious node has the ability to eavesdrop the DH public keys of node A and SN. Besides, the malicious node computes its DH public key P_{Adv} and sends it to node SN and A, respectively. Then, two DH session keys $K_{A \leftrightarrow Adv}$ and $K_{Adv \leftrightarrow SN}$ can be computed which belong to $(A \leftrightarrow Adv)$ and $(Adv \leftrightarrow SN)$ respectively. However, the plaintext message is layer-wise encrypted by each node on the data path and the outermost layer of encryption is calculated using the shared key. As demonstrated in Sect. 5.1.3, the malicious node cannot obtain the shared key. Therefore, it is infeasible for the malicious node to decrypt the ciphertext, thus MITM Attack is effectively defeated.

Table 3 Main initial parameters of simulation scenarios

Parameters	Value
Distribution area	$200m \times 200m$
Number of nodes	100
Location of sink	(50, 50)
Location of other nodes	Randomly generated
Initial energy of node	$1000J$
Initial direct trust value	0.5
Message length	20 bytes
λ_1,λ_2	0.5, 0.5

5.2 Simulation results

This paper uses Matlab for network simulation, and the initial parameters of simulation are set as shown in Table 3. Figure 6 shows the distribution of network nodes. The red node represents the unique sink node, and the remaining nodes represent the other sensor nodes.

Figure 7 is the curve of the Energy Trust value ET of the nodes varying with the traffic W. At the beginning of network operation (i.e. traffic $W = 0$), the ET of all nodes

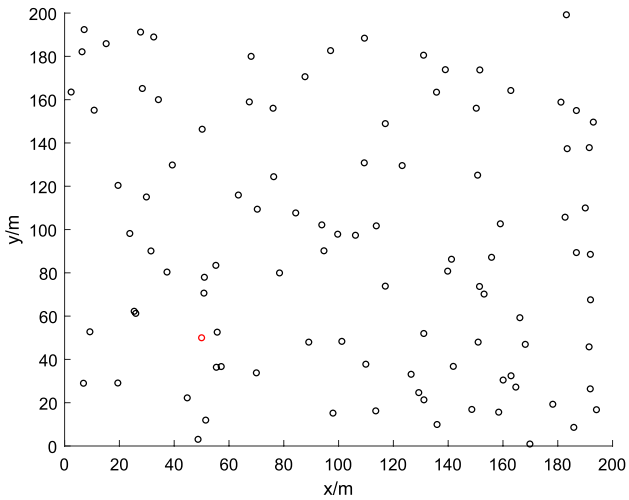

Fig. 6 Network node distribution diagram

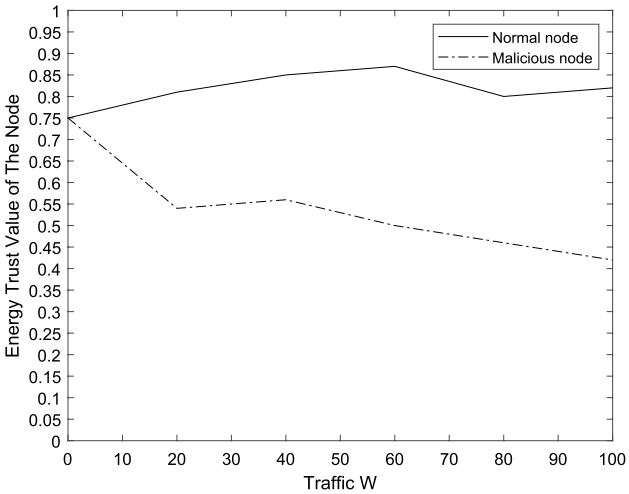

Fig. 7 Energy trust value changes with traffic

is 0.75. The reason is that the initial Direct Trust value of each node is 0.5, and the ratio of remaining energy to initial energy is 1 (i.e. Energy Specification value $ES = 1$). According to the Eq. 10, $ET(W = 0) = 0.5 \times 0.5 + 0.5 \times 1 = 0.75$. With the increase of traffic, the *ET* of normal nodes and malicious nodes is significantly different. In general, the *ET* of normal nodes increases with the increase of traffic, while that of malicious nodes decreases with the increase of traffic.

For normal nodes, when the traffic *W* is between 0 and 60, the number of successful interactions of normal nodes accounts for a higher proportion of the traffic, and its remaining energy is more. Therefore, the *ET* of normal nodes increases with the increase of traffic. When the traffic *W* is between 60 and 80, the *ET* decreases due to the decrease of the remaining energy. At this time, in order to avoid premature depletion of energy, the traffic of normal nodes with low *ET* will be reduced. When the traffic *W* is between 80 and 100, the Direct Trust value increases significantly due to the large increase of the interaction times, so the *ET* of the normal node picks up. However, a large number of communication times bring rapid consumption of node energy. It can be inferred that when the traffic is greater than 100, the *ET* of the node will decrease.

For malicious nodes, when the traffic *W* is between 0 and 20, the *ET* of the node will be greatly reduced due to the large number of failed interactions caused by the malicious behaviors of the node. When the traffic *W* is between 20 and 40, the *ET* of malicious nodes is improved since the remaining energy of malicious nodes is slightly higher than that of normal nodes. However, with the further increase of traffic (i.e. *W* 40), the number of failed interactions of malicious nodes accounts for an increasing proportion of traffic, and its remaining energy is getting less. Therefore, the *ET* of malicious nodes gradually decreases. Therefore, when selecting a node to establish a path according to the *ET*, it can effectively distinguish the malicious node from the normal node and reduce the risk of selecting the malicious node.

Table 4 compares the ETM proposed in this paper with the other three trust models. It can be seen that the four trust models can effectively exclude malicious nodes, but GTRFM (Sinha and Jagannatham 2014), BTMS (Fang et al. 2015), and ADTMS (Luo et al. 2016) all have large energy costs. The model proposed in this paper uses a simplified version of the Beta model and only considers the direct trust value and the remaining energy of the node. Therefore, the energy overhead and computational complexity are within the acceptable range of WSNs.

Figure 8 shows the variation of the average remaining energy of the network with the running time under the DD, ISDD, and TSDDR protocols. The running time is 0, 10, 20, 30, 40, 50, 60, 70, 80, respectively, in seconds. In the initial operation of the network, the energy consumption of the nodes of TSDDR protocol and ISDD protocol is faster, and the average remaining energy of the network is lower than that of the DD protocol. The reason is that both TSDDR protocol and ISDD protocol use cryptography to encrypt plaintext data, which increases the computational complexity with the improvement of security and anonymity, so the nodes consume more energy. Moreover, the computational complexity of TSDDR protocol with the ETM is higher than that of ISDD protocol without the ETM. Therefore,

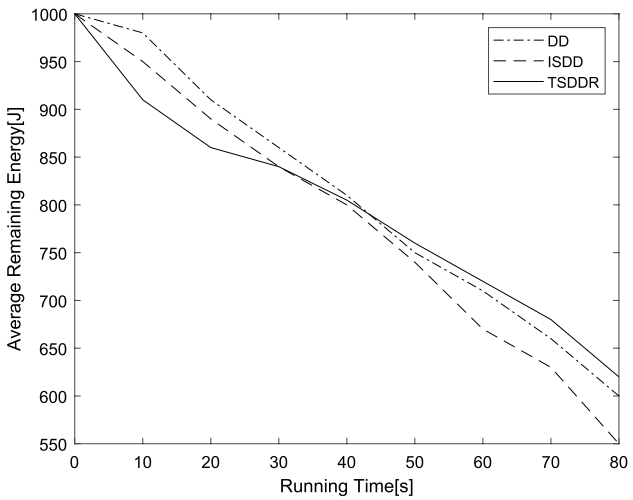

Fig. 8 The average remaining energy of the network with the running time

Table 4 Trust model comparison table

Trust model	Method	Advantage	Limitation
GTRFM	Multivariate Gaussian distribution and Bayesian Trust Model	Effectively isolate malicious nodes	Large energy overhead and high computational complexity
BTMS	Binomial distribution	Effectively resist Collusion Attacks and Slander Attacks	No consideration of energy consumption and weight setting
ADTMS	Hash Algorithm and Beta Density Function Trust Model	Effectively resist External and Internal Attacks	Large energy overhead and high computational complexity
ETM	Simplified Beta Trust Model	Effectively exclude malicious nodes; Extend the life of the nodes	No consideration of indirect trust of nodes

the average remaining energy of the network with TSDDR protocol is lower than that with ISDD protocol. However, with the extension of the running time, the average remaining energy of the network with TSDDR protocol is higher than the other two protocols. This is because the ETM takes into account the Direct Trust value and the remaining energy of a node. When the remaining energy of a normal node decreases, it will have a negative impact on its Energy Trust value, and the traffic of the node will be reduced accordingly. It can be seen that the normal nodes with high Direct Trust values will not be used frequently, thus avoiding the phenomenon of the network hole caused by the premature energy depletion of a single node. This makes the average remaining energy of the whole network in dynamic regulation. Therefore, the TSDDR protocol is suitable for a long-term running detection environment.

6 Conclusion

This paper designs an Energy-based Trust Model and applies it to the Directed Diffusion protocol in WSNs, and proposes a Trust-based Secure Directed Diffusion Routing protocol (TSDDR) to further improve the security and reliability of the data transmission. In addition, the TSDDR protocol uses IBC, DH key exchange protocol, and Bilinear Pairing to protect the confidentiality of data and the anonymity of nodes. Security analysis shows that the proposed protocol can not only achieve anonymous communication between nodes, end-to-end data security, but also prevent external malicious nodes from impersonating legitimate nodes on the path and launching man-in-the-middle attacks. Simulation results show that the proposed protocol can effectively eliminate malicious nodes when selecting relay nodes to establish paths. Moreover, the average remaining energy, that is, the life cycle of the network is also increased.

Acknowledgements This work was partly funded by EU Horizon 2020 DOMINOES Project (Grant Number: 771066) and CERNET Innovation Project (NGII20181201)

References

Dai S, Tang J, Zhang A (2010) Gossiping-based directed diffusion for wireless sensor network. Inf Secur Commun Priv 4:1418–1430

El Mougy A, Sameh S (2018) Preserving privacy in wireless sensor networks using onion routing. 2018 international symposium on networks. computers and communications (ISNCC), IEEE, pp 1–6

Fang W, Zhang X, Shi Z, Sun Y, Shan L (2015) Binomial-based trust management system in wireless sensor networks. Chin J Sens Actuators 28(5):703–708

Fei S, Yu G, Hao Z (2007) Secure directed diffusion protocol based on random key predistribution model. J Xian Jiaotong Univ 41(12):1423

Feng L, Li Q, Zhang M (2014) Improvement in spin protocol based on dynamic routing. J Dali Univ 12:8

Hiller J, Pennekamp J, Dahlmanns M, Henze M, Panchenko A, Wehrle K (2019) Tailoring onion routing to the internet of things: Security and privacy in untrusted environments. In: 2019 IEEE 27th international conference on network protocols (ICNP), IEEE, pp 1–12

Jiang N, Xu D, Zhou J, Yan H, Wan T, Zheng J (2020) Toward optimal participant decisions with voting-based incentive model for crowd sensing. Inf Sci 512:1–17

Li Q, Meng X, Huang L (2014) VPN technology realization and network influence analysis based on Diffie-Hellman algorithm. Software Engineer

Li J, Wang X, Huang Z, Wang L, Xiang Y (2019) Multi-level multi-secret sharing scheme for decentralized e-voting in cloud computing. J Parallel Distrib Comput 130:91–97

Liu Z, Wang L, Wang X, Shen X, Li L (2019) Secure remote sensing image registration based on compressed sensing in cloud setting. IEEE Access 7:36516–36526

Luo W, Ma W, Gao Q (2016) A dynamic trust management system for wireless sensor networks. Secur Commun Netw 9(7):613–621

Ren B, Liu L, Ma J (2006) A novel directed diffusion mechanism for wireless sensor networks. J Electron Inf Technol 28(3):562–568

Roy S, Das AK (2014) Secure hierarchical routing protocol (SHRP) for wireless sensor network. In: International symposium on security in computing and communication, Springer, pp 20–29

Sengupta J, Ruj S, Das Bit S (2018) An efficient and secure directed diffusion in industrial wireless sensor networks. In: Proceedings of the 1st international workshop on future industrial communication networks, pp 41–46

Sengupta J, Ruj S, Bit SD (2019) End to end secure anonymous communication for secure directed diffusion in IoT. In: Proceedings of the 20th international conference on distributed computing and networking, pp 445–450

Sinha RK, Jagannatham AK (2014) Gaussian trust and reputation for fading mimo wireless sensor networks. 2014 IEEE international conference on electronics, computing and communication technologies (CONECCT), IEEE, pp 1–6

Wang X, Zhang Y, Gupta BB, Zhu H, Liu D (2019) An identity-based signcryption on lattice without trapdoor. J UCS 25(3):282–293

Ye J, Yang J, Song X (2012) Cross-layer congestion control approach based on directed diffusion routing protocol in WSN. Chin J Sens Actuators 1:1–16

Ye Z, Wen T, Liu Z, Fu C (2019) An algorithm of trust-based secure data aggregation for wireless sensor networks. J Northeastern Univ 12:98–110

Zhang F, Safavi Naini R, Susilo W (2004) An efficient signature scheme from bilinear pairings and its applications. In: International workshop on public key cryptography, Springer, pp 277–290

Zhao S, Aggarwal A, Frost R, Bai X (2012) A survey of applications of identity-based cryptography in mobile ad-hoc networks. IEEE Commun Surv Tutor 14(2):380–400

Zheng M, Li Y, Zhao XC, Zhou H (2013) Behavior simulation for wireless sensor networks based on directed diffusion. Comput Syst Appl