Lightweight and Secure Data Transmission Scheme Against Malicious Nodes in Heterogeneous Wireless Sensor Networks

Na Wang, Shancheng Zhang, Zheng Zhang, Jiawen Qiao, Junsong Fu, Jianwei Liu, *Senior Member, IEEE*, and Bharat K. Bhargava, *Life Fellow, IEEE*

***Abstract*— With the continuous development of sensor technology, more and more users hope to monitor and collect information in a certain area safely and efficiently by deploying heterogeneous wireless sensor networks (HWSNs). However, nodes in HWSNs have limited capabilities, which leads to many security challenges. Existing data transmission schemes in HWSNs take measures to resist these security threats, which aggravate the node computation overhead and increase the network energy consumption. This paper proposes a Lightweight and Secure Data Transmission (LSDT) scheme against malicious nodes in heterogeneous wireless sensor networks. Firstly, considering node capabilities limitations in HWSNs, we design a lightweight secret sharing scheme based on XOR operation, which maps data to multiple shares and makes it convenient to transmit shares separately to the sink node via multiple paths. While guaranteeing data security, this scheme can greatly reduce the computation overhead of nodes compared with traditional secret sharing schemes. Further, during the delivery of shares, the network may be attacked by malicious nodes, causing the interruption of message transmission. Therefore, we design a malicious node detection and feedback mechanism, which can quickly respond to malicious node attacks and update the reputation degree of malicious nodes. Finally, we propose a routing selection scheme based on reference path which comprehensively considers the energy and reputation degree of heterogeneous nodes. It makes message transmission bypass malicious nodes while achieving network energy load balance, significantly extending the network lifetime. The security analysis proves that our scheme guarantees the security of data transmission. Theoretical analysis and experiments show that our scheme has significant advantages over the existing HWSNs data transmission schemes in terms of network lifetime extension and malicious node resistance.**

***Index Terms*— Heterogeneous wireless sensor networks, load balance, lightweight secret sharing, secure data transmission.**

I. Introduction

With the continuous progress of technology, small and cheap wireless sensors are widely deployed, providing solid support for the development of wireless Internet of Things (IoT). Wireless sensor networks (WSNs) have become one of the most potential technologies in the IoT technology, playing an important role in such fields as industrial detection [1], intelligent city environmental monitoring [2], wildlife monitoring [3] and so on. According to the similarities and differences of node structure, energy, function and link, WSNs can be divided into homogeneous wireless sensor networks and heterogeneous wireless sensor networks (HWSNs) [4]. However, compact and inexpensive sensor nodes have many limitations, such as small energy reserves, weak computing and storage capabilities, short-distance communication ability, and node updating difficulty [5]. This means that the sensor node cannot interact directly with the terminal server and must conduct multi-hop routing transmission through relay sensor nodes. In addition, due to a large number of nodes in HWSNs and the difficulty to monitor the deployment locations all the time, the adversary is prone to corrupt some of the nodes and thus damage the availability of the network, such as tampering or deleting the transmitting messages [6]. Therefore, data transmission in HWSNs faces many challenges.

Due to the limited resources of wireless sensors, how to extend the network lifetime is the first problem to be considered. Researchers' ideas are mainly divided into two categories. One is to optimize the energy loss of data transmission by designing lightweight data processing schemes or reducing the transmitted messages [7], [8]. Another starts from energy load balancing. Through dynamic route selection, they balance the energy consumption of each node to avoid the impact of the energy loss of some nodes on the overall network lifetime [9], [10]. The representative of the first type is scheme in [8], which reduces the node communication overhead and improves energy efficiency by transmitting the key index instead of the key itself. In the second type of solutions to network energy load balancing, the environment fusion multi-path routing protocol (EFMRP) proposed by Fu et al. [10] is relatively typical. The basic idea of this method is to trade off communication delay, energy consumption and route lifetime to get the best route decision. Although the above schemes can prolong the network lifetime, the security of data transmission is not considered enough.

Manuscript received 30 September 2022; revised 25 April 2023 and 17 June 2023; accepted 12 July 2023. Date of publication 21 July 2023; date of current version 2 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2702700, in part by the National Natural Science Foundation of China under Grant 62102017, in part by the Beijing Natural Science Foundation under Grant L222050 and Grant M22038, and in part by the Fundamental Research Funds for the Central Universities under Grant YWF-23-L-1240. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Andrew Clark. *(Corresponding author: Junsong Fu.)*

Na Wang, Shancheng Zhang, Zheng Zhang, Jiawen Qiao, and Jianwei Liu are with the School of Cyber Science and Technology, Beihang University, Beijing 100191, China (e-mail: nawang@buaa.edu.cn; zscbuaa@buaa.edu.cn; ZZ_shiwo@buaa.edu.cn; selina@buaa.edu.cn; liujianwei@buaa.edu.cn).

Junsong Fu is with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: fujs@bupt.edu.cn).

Bharat K. Bhargava is with the Department of Computer Science, Purdue University, West Lafayette, IN 47906 USA (e-mail: bbshail@purdue.edu).

Digital Object Identifier 10.1109/TIFS.2023.3297904

HWSNs is vulnerable to attacks because of its open wireless communication pattern and great management difficulty [11]. Common attacks include eavesdropping and tampering attack [12]. The multi-path routing algorithm has the ability to mitigate the impact of malicious nodes on data transmission routes, thus ensuring the successful delivery of data to the intended sink node. Despite its robustness against tampering and other forms of attacks, the algorithm is not immune to eavesdropping attacks, whereby adversaries can intercept and access transmitted data. This problem can be effectively solved by introducing encryption and secret sharing [13], [14] into the multi-path routing scheme. After the data is encrypted, it is divided into multiple shares and transmitted separately in the established multiple routes. The original data is finally recovered by the sink node. Thus, under the (t, n)-threshold, the adversary needs to intercept at least t shares in order to eavesdrop on the encrypted message. Therefore, the multi-path routing protocol based on secret sharing greatly improves the fault tolerance rate of data transmission. Meanwhile, the data confidentiality and integrity are guaranteed.

In addition, in recent HWSNs attack schemes, researchers pay more attention to route attack [15], such as wormhole attack, black hole attack, grey hole attack, deception attack, Sybil attack and so on. Nodes corrupted by adversary are said to be malicious nodes. Taking black hole attack as an example, malicious nodes claim that they have great transmission advantages, attracting surrounding nodes to transmit data to them. However, they do not forward the data, thus destroying the availability of the network within the communication range. The existing schemes to resist malicious node attacks mainly include three categories [11], which are based on anomaly [16], feature [17] and general intrusion detection extension scheme [18] respectively. While the aforementioned schemes employ diverse mechanisms to safeguard against malicious node attacks and ensure secure data transmission in HWSNs, they still suffer from issues such as limited detection efficiency and slow response speed [12].

To solve these problems, we propose a Lightweight and Secure Data Transmission (LSDT) through multi-path routing based on XOR operation for HWSNs. Specifically, we use the XOR operation to encrypt ciphertext data after splitting it, quickly generate multiple shares, and then transmit them to the sink node through multi-path routing. The sink node only needs to obtain part of shares to recover the complete original data. In addition, the routing function is designed properly so that the relay node can consider both load balancing and resisting malicious nodes when choosing the routing path for message transmission. Thus it ensures the security of message transmission and prolongs the network lifetime. Further, in order to quickly detect malicious behaviors and locate malicious nodes, we introduce a dynamic malicious node feedback management mechanism. Finally, the security analysis proves that our LSDT scheme effectively protects the integrity, confidentiality and availability of transmitted messages under malicious node attacks. The theoretical analysis and experimental results show that compared with other similar schemes, LSDT significantly reduces the node calculation cost, balances the energy consumption of each node and prolongs the service lifetime of the network. At the same time, the malicious node resistance mechanism effectively prevents attacks, thus avoiding the influence of malicious nodes and guaranteeing the network availability.

The innovations and contributions of our scheme are summarized as follows:

- We design a lightweight and secure multi-path routing data transmission scheme for HWSNs. We innovatively propose a threshold secret sharing technology based on XOR operation. In this way, the node computation cost is reduced and the sink node can recover the original data despite partial loss.
- In order to balance the energy consumption of each part of the network, extend the network lifetime, and resist malicious nodes, we design a real-time decision routing scheme. The scheme chooses the optimal path to transmit messages by considering the information of node energy, malicious nodes and transmission path length.
- As for malicious node attack, we also propose an efficient malicious node management mechanism. This mechanism enables fast localization of malicious nodes and make the message transmission path bypass malicious nodes, improving the robustness of the system.
- We conduct a comprehensive analysis of the scheme from the perspectives of theory and experimental simulation. The results show that, compared with similar schemes, the LSDT scheme significantly prolongs the network lifetime and ensures the transmitted data security.

The rest of the paper is organized as follows. Section II reviews the previous related works. Section III shows the preliminaries. Section IV describes the system model and problem statement. Section V elaborates on the main design of the LSDT scheme. The security analysis in theory for the LSDT scheme is provided in Section VI. Section VII gives the performance of the LSDT scheme. Finally, the conclusion and future work are discussed in Section VIII.

II. Related Work

In HWSNs, conventional data transmission schemes often transmit the data collected by sensor nodes to the sink node through fixed routing paths. However, due to the limited energy, the heavily loaded nodes will fail prematurely, making the routing through the node invalid. Accordingly, the data collected by the corresponding sensor nodes will be lost [19]. An intuitive solution is to establish multiple feasible routing paths to improve the robustness of data transmission and meet the load balancing requirement of each node. In the earliest multi-dataflow topologies algorithm [20], data is transmitted through two routing paths at the same time. Even if there is a malicious node on one routing, the data can be normally transmitted to the sink node. Later multi-path routing schemes [21], [22], [23] were optimized for energy consumption and load balancing. Sajwan et al. [21] proposed an algorithm that maximizes energy efficiency using planar and hierarchical routing schemes. They used a multi-hop routing scheme and cluster head communication to reduce energy consumption in the network. Sakthidasan et al. [22] used the ant colony algorithm to select the path with high reliability based on fuzzy logic according to link stability, residual energy and packet loss rate. Jemili et al. [23] proposed a cross-layer multi-path routing approach considering different context information. In this way, node-disjoint paths are established to concurrently transport multimedia content from sources to the sink, which minimizes energy consumption and extends network lifetime. These schemes [21], [22], [23] provide a variety of feasible multi-path routing scheme design ideas and extend the network

lifetime by designing lightweight algorithms. But they do not guarantee the security of the transmitted data.

In order to ensure the confidentiality and integrity of data during transmission, Lou et al. [24] introduced a secret sharing mechanism into the multi-path routing scheme and proposed the hybrid multi-path scheme. Then, the end-to-end data transmission security in this scheme is enhanced. After that, Deryabin et al. [25] used the secret sharing scheme based on Residue Number System (RNS), which contributes to solving the problem of confidentiality and integrity of transmitted data. In addition, reducing the energy consumption of share-based multi-path routing schemes is also a concern of researchers. Chen et al. [14] designed a lightweight secret sharing scheme and an appropriate random routing algorithm to transmit shares to the sink node through multiple paths. Simulation results show that this method effectively reduces network energy consumption and resists eavesdropping and backtracking attacks. Haseeb et al. [26] proposed an energy-aware and multi-hop routing protocol by using equal-amount secret sharing scheme based on XOR operations. That improves energy efficiency and data security against malicious behaviors.

The above schemes [24], [25], [26] can resist eavesdropping and tampering attacks in the process of data transmission. Unfortunately, these schemes do not take the routing attacks of HWSNs into account, causing low system availability. Jahandoust et al. [18] proposed a distributed adaptive framework based on the subjective logic and probabilistic extension of timed automata. It captures the behavior of the entire network to deduce the probability that each node is affected by black hole attacks. Merlin et al. [27] proposed a trust-based energy-aware routing mechanism to detect black hole attacks as quickly as possible by dynamically generating multiple detection paths. At the same time, it obtains node trust to provide better data routing security. Moreover, comprehensively considering energy consumption and security issues, [28], [29] proposed lightweight schemes to resist black hole attacks. The lightweight trust-enhanced ad hoc on-demand multi-path distance vector protocol proposed in [28] only uses passive and local monitoring information to evaluate the behavior of entities. Thus it achieves the effect of lightweight black hole attack resistance. Liu et al. [29] presented an active detection-based secure and trust routing scheme named ActiveTrust. This scheme significantly improves the probability of successful data transmission and the ability to resist black hole attacks, as well as prolonging the network lifetime.

In order to further improve the robustness of the system, a feasible idea is comprehensively considering secret sharing and attack resistance to design a multi-path routing scheme. Shu et al. [30] designed a mechanism for generating random multi-path routes where the routes adopted for the shares of different packets change over time. Besides, the resulting paths are highly decentralized and energy efficient, allowing them to bypass the black hole. Liu et al. [31] described the multi-path routing problem based on secret sharing as an optimization problem. The goal is to maximize network security and longevity in the case of energy constraints. Both theoretical and simulation results show that the scheme can significantly improve network security with single black hole and multiple black holes.

To sum up, the proposed schemes have solved part of the security and energy consumption problems in HWSNs. However, it is challenging to achieve a proper balance of network load and lightweight data transmission while mitigating the effects of malicious nodes. To this end, we design a lightweight and secure multi-path routing data transmission scheme for HWSNs.

III. Preliminaries

A. Encryption Schemes

Encryption schemes can be divided into symmetric encryption schemes (e.g. AES) and asymmetric encryption schemes (e.g. ECC). For the convenience of expression, the cryptographic primitives involved in the system are expressed in formal language as follows.

1) Symmetric Encryption Scheme $\mathcal{SE}$: In our system, the symmetric encryption scheme $\mathcal{SE}$ is used to encrypt data during data transmission. $\mathcal{SE}$ consists of three algorithms:

- $\mathcal{SE}.Setup(1^\kappa) \rightarrow key$: Input the security parameter κ to generate a random symmetric key key.
- $\mathcal{SE}.Enc(dat, key) \rightarrow dat_E$: Input the plaintext dat and symmetric key key, and output the ciphertext dat_E.
- $\mathcal{SE}.Dec(dat_E, key) \rightarrow dat$: Input the ciphertext dat_E and symmetric key key, and output the plaintext dat.

2) Asymmetric Encryption Scheme $\mathcal{AE}$: In our system, the asymmetric encryption encryption scheme $\mathcal{AE}$ is used to encrypt the session key. $\mathcal{AE}$ includes three algorithms:

- $\mathcal{AE}.Setup(1^\kappa) \rightarrow (pk, sk)$: Input the security parameter κ to generate a random public/private key pair (pk, sk).
- $\mathcal{AE}.Enc(key, pk) \rightarrow key_E$: Input session key key and public key pk, and output session key ciphertext key_E.
- $\mathcal{AE}.Dec(key_E, sk) \rightarrow key$: Input session key ciphertext key_E and private key sk, and output session key key.

B. Hash Function

As a common cryptographic tool, hash is widely used in integrity verification, encryption, digital signature and other problems [32], [33]. Our scheme uses the hash function to provide integrity verification of the transmitted data. Compared with the scheme based on random oracle [33], the hash function used in our scheme only needs to satisfy the collision resistance. For the convenience of presentation, this subsection gives the definition of this property:

Definition 1 (Collision Resistance): Hash function H is collision resistance if and only if for any message m, there are different messages $m' \neq m$ such that the following equation holds.

$$\Pr[H(m') = H(m)] < \epsilon, \tag{1}$$

where ϵ is a function whose value is negligible.

C. Secret Sharing

The concept of secret sharing was suggested by Shamir [34] in 1979. It is necessary to design a lightweight scheme in order to meet the requirement of low energy load in HWSNs. (t, n)-threshold secret sharing schemes generally contain two algorithms: secret distribution and message recovery.

- Secret distribution. For the input message m, n shares $S = \{s_i\}_{1 \leq i \leq n}$ are generated.
- Message recovery. Take at least t shares $S' \subset S$, $|S'| \geq t$ as input, and output m.

Its security is defined for subsequent security analysis.

Definition 2: A (t, n)-threshold secret sharing scheme is secure if and only if, for a probability polynomial time (PPT) adversary $\mathcal{A}$, the following formula holds:

$$\Pr\left[m = \mathcal{A}\left(S'\right)\right] < \epsilon, S' \subset S, \left|S'\right| < t, \tag{2}$$

where ϵ is a function of negligible value.

IV. Problem Formulation

This section describes the data transmission system model of HWSNs and lays the foundation for the design of the LSDT scheme. First, we give the LSDT system model in Section IV-A, detailing the heterogeneous sensor nodes and the specific process of message transmission in the network. Then, we introduce the network lifetime model and security model in the HWSNs in sections IV-B and IV-C, respectively. Finally, in Section IV-D, we give the design goals of the LSDT scheme and verify that this scheme achieves the design goals in the later scheme analysis and simulation experiments.

A. The System Model

Consider a data transmission scenario in heterogeneous wireless sensor networks (HWSNs) in the real world, as shown in the concrete model in Fig. 1. Sensors with different entities in a certain area would collect data and transmit messages to a base station. The base station then transmits messages to the Internet. There may be malicious sensors trying to disrupt the transmission of messages. The abstract model in Fig. 1 is obtained by abstracting the entity in the concrete model. Among them, heterogeneous sensor nodes are randomly deployed in the sensing field and divided into two roles: sensor nodes and the sink node. Data transmission in the system consists of two processes: internal message transmission and external network communication. Sensor nodes (i.e., source nodes) collect data nearby and generate messages. Then they establish routing paths to the sink node, and transmit messages to the sink node through relay nodes. Finally, the sink node processes the messages and transmits the original data to the user through the external network.

Sensor nodes. A group of heterogeneous sensor nodes is deployed in the sensing area as shown in Fig. 1. Among them, the different shapes of nodes represent their heterogeneity, and the difference in color represents the difference in residual energy of nodes. For example, green means energy abundance, while red means energy scarcity. Each sensor node u has initial energy E_u and fixed transmission radius D_u. Also, it has unique identity ID_u and fixed location coordinate $Lo_u = (x_u, y_u)$. The node u has the ability to broadcast radio messages in D_u and know the distance information of neighbor nodes in D_u. They mainly perform the functions of collecting data in the D_u area and transmitting messages to neighbor nodes. It should be noted that heterogeneous nodes have different initial energy and transmission radii.

The sink node. In HWSNs, the sink node s is responsible for interacting with the remote Internet and has fixed position coordinate $Lo_s = (x_s, y_s)$. In the network initialization phase, all nodes know the location coordinate of the sink node. The sink node has high energy and the ability to broadcast to the whole network. It is mainly responsible for receiving and summarizing the data collected in the network and managing the whole network. The sink node has the highest authority of HWSNs and manages sensor nodes, while sensor nodes cannot affect other nodes.

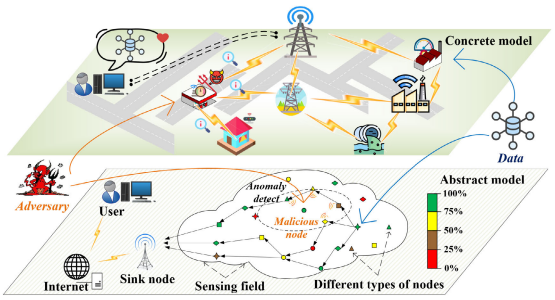


Fig. 1. The system model.

Internal message transmission. In the network initialization phase, each node needs to establish a routing table T_u leading to the sink node according to the routing paths initialization algorithm. The routing table stores the routing paths through relay nodes to the sink node. Sensor node u sends the message to the sink node by forwarding it through multiple hops according to the reference paths in T_u. Here, each sensor node plays two roles. As a source node, the sensor node collects data and processes it to generate messages, which are then transmitted to neighbor nodes. As a relay node, it must select an appropriate next-hop node in order to forward the message to the sink node.

External data transmission. The sink node has the ability of the whole network broadcast and it can interact with the external Internet. The user obtains the data collected by the whole HWSNs through the external Internet.

B. Network Lifetime Model

1) Energy Consumption Model: In this paper, the commonly used energy consumption model [35] will be introduced to analyze the energy consumption of wireless transmission. Suppose that the energy consumption of node u sending message to node v is expressed as follows:

$$E_T(d_{uv}, L) = \begin{cases} LE_{elce} + L\xi_{fs}d_{uv}^2, d_{uv} \le D_0; \\ LE_{elce} + L\xi_{mp}d_{uv}^4, d_{uv} > D_0. \end{cases} \tag{3}$$

where d_{uv} represents the distance between nodes u and v. And the energy consumed by node v to receive data from node u is calculated as follows:

$$E_R(d_{uv}, L) = LE_{elce}, \tag{4}$$

where L is the message bit length. E_{elce} is the unit energy consumption coefficient. ξ_{fs},ξ_{mp} represent the power amplifier parameters under the free space model and the multi-path attenuation model, respectively. D_0 is the distance threshold.

2) Network Lifetime: To quantitatively analyze the lifetime of HWSNs, we configure all source node within the network to transmit data collected during a single cycle to the sink nodes in a single round of data transmission. We hereby define the package delivery rate (PDR) as $\mathfrak{r}$:

$$\mathfrak{r} = \frac{\mathfrak{m}}{\mathfrak{M}}, \tag{5}$$

where $\mathfrak{m}$ denotes the amount of data successfully received and recovered by the sink node while $\mathfrak{M}$ denotes the total amount of data sent by all nodes per round. PDR $\mathfrak{r}$ effectively reflects the network availability situation. Without considering the influence of malicious nodes, if the $\mathfrak{r}$ is low during a round of data transmission, it indicates that the network is affected

by low energy nodes and cannot complete data collection work. For this reason, we define the *network lifetime* as the maximum number of data transmission rounds that maintain $\mathfrak{r} > 50\%$ starting from the deployment of the network.

C. Security Model

1) Threat Model: In HWSNs, there is a finite number of malicious nodes. Each malicious node d has extremely large energy and limited communication range D_d and is included in the routing table of nearby sensor nodes. Considering the process that the source node sends message Meg including dat to the sink node through the path in routing table T_s, the behavior patterns of the malicious node d are shown as follows:

- When the malicious node is a relay node forwarding message Meg_i, it modifies $Meg_i' \neq Meg_i$ and forwards to the next-hop node v_j.
- The malicious node d will listen to all messages $\{Meg_i\}$ in the D_d range, and look for a protocol algorithm $\mathcal{B}$: $\mathcal{B}(\{Meg_i\}) \rightarrow dat$ to crack the plaintext data dat.
- The malicious node d declares its own energy $E_d = \infty$ to other nodes v_j in the D_d range, so that $IF(v_j, d)$ has a significant advantage when the neighbor node v_j calculates the relay node selection parameter IF. In this way, all nodes in the D_d range will forward messages to d. d may selectively forward some packets [36] or not forward [37].

2) Security Goals: As a data transmission and communication scheme, we focus on data security. According to the standard CIA definition [38] of data security, data security includes three properties: integrity, confidentiality and availability. Under our threat model defined in Section IV-C, all three security properties will be compromised. For example, malicious nodes tamper with the forwarded message, which disrupts data integrity; Malicious nodes attempt to access the original data dat, which destroys confidentiality; Malicious nodes withhold forwarding message, which compromises data availability. Therefore, our security goals are to resist the threat of malicious nodes to achieve CIA security. The security goals of the solution are formally defined as follows:

Definition 3: A PPT adversary $\mathcal{A}$ is set to tamper the message m into $m' \neq m$. Then a scheme is called data integrity protected if and only if

$$\Pr\left[H\left(m'\right) = H(m)\right] < \epsilon, \tag{6}$$

where ϵ is a function with negligible value.

Definition 4: For any PPT adversary $\mathcal{A}$, if the following formula holds, the data in the scheme is assumed to satisfy confidentiality:

$$\Pr\left[dat \leftarrow \mathcal{A}\left(\{(i, s_i)\}\right)\right] < \epsilon, \tag{7}$$

where ϵ is a function whose value is negligible.

Definition 5: Ideally (sufficient energy, etc.), in HWSNs with finite malicious nodes, for the set $Message = \{m_i\}$ composed of $\mathcal{N}$ message m_i sent by source nodes in network, messages received by the sink node is denoted as $Message' = \{m_i'\}$ after being transmitted. If the following formula is true, then the scheme meets the data availability:

$$\forall i \in [\mathcal{N}], \ \frac{\sum \Pr([m_i \neq m_i'])}{\mathcal{N}} < \epsilon, \tag{8}$$

where ϵ is a function of negligible value.

D. Design Goals

In order to achieve a lightweight and secure transmission scheme in HWSNs, our scheme LSDT should meet the following design goals.

- *Low computational overhead.* Due to the limited energy of sensor nodes, data generation and procession should be lightweight and efficient to reduce the computational overhead of source nodes.
- *Load balancing.* Considering the energy constrained HWSNs, the designed data transmission scheme should be load balanced, which can comprehensively think about the energy consumption of each node in the network and prolong the network lifetime.
- *Data integrity and confidentiality.* The designed transmission scheme should protect the integrity and confidentiality of the data sent by the source node and transmit the messages to the sink node.
- *Resisting malicious nodes.* The proposed data transmission scheme should be able to resist malicious node attacks and maintain network availability.

V. Construction of LSDT System

Given the relevant model in the previous section, as shown in Fig. 2, our scheme LSDT is divided into four processes in order of operation, which describe the initialization of the network, data encryption and shares generation, message transmission, and original data recovery. In this section, a lightweight secret sharing algorithm and malicious node management mechanism are designed.

A. Initialization

The initialization of the network requires the participation of all nodes. This process is divided into two steps. First, during the deployment of the HWSNs, public parameters such as the public key of the whole network and the location of the sink node, and private parameters such as the key, energy and detection radius of each node need to be generated one by one. Then, in order to connect to the sink node, each node needs to establish the initial routing path to ensure that all sensor nodes can receive the public parameters broadcast by the sink node. In the following, we give the detailed structure of public parameters, node initialization information and then propose the maximum P-hop routing broadcast construction algorithm of this scheme. In this way, each node in the network obtains the necessary information and the whole network is connected. The initialization process completes.

1) Initialization of Parameters: Suppose that HWSNs is static networks in a two-dimensional plane area, and N nodes are randomly deployed in the target monitoring area with the size of $W \times W$. As mentioned in the system model in Section IV-A, nodes are divided into two categories, namely, heterogeneous sensor nodes and the sink node. For heterogeneous sensor nodes, we use a set of parameters to characterize their capabilities, while the sink node is responsible for generating public parameters of the entire network and broadcasting them to each node.

Since the network consists of a group of heterogeneous sensor nodes, the initial energy and communication radius of each node are different. We assume that heterogeneous nodes are equipped with different initial energy values in the interval $[E_0, (1+\theta)E_0]$, where E_0 is a basic unit of energy, θ is a coefficient variable and $\theta > 0$. Each node u

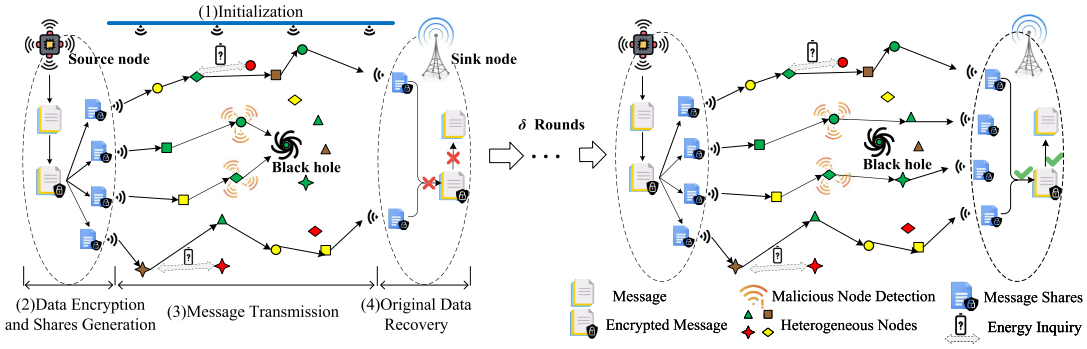


Fig. 2. Process flow of LSDT. (Take (3,4) secret sharing scheme as an example. In the left subfigure, only two routes successfully reached the sink node due to the black hole attack carried out by the malicious node, and the first round of data transmission failed. After δ rounds data transmission, as shown in the right subfigure, the network can evade the malicious node and successfully recover the message).

can communicate with neighbor nodes in the range of D_u. It should be noted that for two heterogeneous neighbor nodes with different communication radii, the distance between them must be less than the communication radius of either node.

Parameters initialization is performed by the sink node during the system deployment phase. The sink node s needs to conduct the following operations:

- Get its location coordinate $Lo_s = (x_s, y_s)$.
- Run the algorithm $\mathcal{AE}.Setup(1^\kappa)$, and generate public/private key pair (pk, sk) for asymmetric encryption.
- Set the value of weighting coefficient of routing selection parameter λ, the integrity authentication hash function H and the calculation function f of the maximum number of hops P.
- Construct the multiplicative cyclic group G_q, q is a large prime number, and $g \in G_q$ is a generator.
- Generate initialization information

$$PPK = \left\{Lo_s, pk, f, \lambda, H, G_q, g\right\}, \tag{9}$$

$$SM = \{PPK, hop = 0, R = \{\}\}, \tag{10}$$

where hop is the hop count recorder of the SM when the network executes the broadcast initialization, and R is a list to store the routing path, which is initially empty.

2) Reference Routing Path Initialization: After the public parameters are generated, the sink node needs to broadcast them to the whole network, which needs to construct the initial routing paths from all nodes to the sink node. In order to adapt to a multi-path routing scheme, we design a maximum P-hop routing broadcast. The routing algorithm aims to find all the maximum P-hop route to the sink node for each node, so as to establish multiple transmission routes for the data collected by sensor nodes. During the initialization process, the node needs to build the initial maximum P-hop route in three steps.

First, the sink node broadcasts the network initialization information SM to nodes within the distance D_0. After receiving the initialization information and storing PPK, the node u adds R in SM as a new path to the local storage routing tables T_u and adds its own ID_u to R to generate new initialization information $SM' = \{PPK, hop + 1, R \| ID_u\}$. After that, SM' is sent to the neighbor node v within the distance D_u. The distance between node v and the sink node s should be greater than that between node u and the sink node s, i. e. $d_{vs} > d_{us}$.

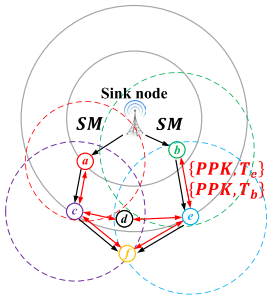
After the first step, for node u that is broadcast to, set n to be the maximum number of hops of the storage routing paths in T_u, and d_{us} to be the physical distance between node u and the sink node s. Set $P = f(n, d_{us}) = n + \lceil \frac{d_{us}}{averd_s} \rceil$ to the maximum number of hops, in which the parameter $averd_s$ is the average distance between the sink node and the nodes within D_0 from the sink node, and it's stored in the PPK along with f. Set $P = NULL$ for nodes that are not broadcast.

After that, the neighbor nodes of the sink node are used as the starting points to construct the maximum P-hop route broadcast as shown in Algorithm 1. Consider the process of node u sharing routing update information $\{PPK, T_u\}$ to its neighbor node v. If P of v is not $NULL$, the path whose route length is less than or equal to $P-1$ is selected from T_u and added to T_v. Otherwise, P and T_v are initialized based on T_u. Specifically, v first stores the public parameter PPK, and then uses the sink node position Lo_s in the PPK to calculate the distance $d_{vs} = \|Lo_s - Lo_v\|_2$ between itself and the sink node. Assuming that the maximum number of hops of the paths in the received routing table T_u is n, then v initializes the maximum number of routing hops $P = f(n + 1, d_{us})$. Finally,

Algorithm 1 Reference routing path initialization

Require: PPK, T_u, T_v, P_v
Ensure: T_v, P_v

```
Node v receives the routing update information PPK, T_u sent by the neighbor node u, and sets change = 0
if P_v == NULL then
    Store the public parameter PPK, set change = 1
    Calculate d_vs = ||Lo_s − Lo_v||_2
    Take the maximum number of hops in the paths in T_u as n
    Calculate P_v = f(n + 1, d_vs)
    Append ID_v to each path of T_u and save them as T_v
else
    if There is a set of paths in T_u that do not belong to T_v and whose routing hops are less than P then
        Append ID_v to each path in the set and update them to T_v
        set change = 1
    end if
end if
if change == 1 or routing update information is received for the first time then
    Send PPK, T_v to v's neighbor nodes except u.
end if
```



ID	T_{ID}		P	
	step 1	step 3	step 2	step 3
a	$\{(a,s)\}$	$\{(a,s)\}$	2	2
b	$\{(b,s)\}$	$\{(b,s)\}$	2	2
c	$\{(c,a,s)\}$	$\{(c,a,s),(c,d,e,b,s),(c,f,e,b,s)\}$	4	4
d	$\emptyset$	$\{(d,c,a,s),(d,e,b,s),(d,f,c,a,s),(d,f,e,b,s)\}$	$NULL$	5
e	$\{(e,b,s)\}$	$\{(e,b,s),(e,d,c,a,s),(e,f,c,a,s)\}$	4	4
f	$\{(f,c,a,s),(f,e,b,s)\}$	$\{(f,c,a,s),(f,e,b,s),(f,d,e,b,s),(f,d,c,a,s),(f,c,d,e,b,s),(f,e,d,c,a,s)\}$	6	6

Fig. 3. The process of constructing HWSNs routing.

v appends ID_v to each path of T_u and saves them as T_v. After the receiving process is completed, if T_v has changed, v will continue to share $\{PPK, T_v\}$ with its neighbors except u.

The initialization ends when no more routing table updates occur in the network. At this point, each node gets a table of short routing paths.

Next, we elaborate on the process of constructing HWSNs routing by broadcasting through an example. As shown in Fig. 3, in addition to the sink node s, the network contains six sensor nodes a, b, c, d, e, f. The dotted circles with different colors represent the signal transmission range D_u ($u \in \{a, b, c, d, e, f\}$) of each heterogeneous node, respectively. The gray solid line circle represents the equidistant line from s. In the first step of initialization, s sends initialization information SM to two neighbor nodes a and b, where the path recorder $R = \{\}$. After that, a and b respectively add the paths (a, s) and (b, s) to the local routing table T, update R to $\{a\}$ or $\{b\}$, and forward the updated initialization information SM' to their respective neighbor nodes c, e. Next, c, e propagate the initialization information to their neighbor node f instead of d. This is because the algorithm requires that the propagation direction is always away from s, and d is closer to s than c and e are. After the first step of initialization, all five nodes except d have obtained public parameters and part of the routing paths to the sink node. The second step is to calculate the maximum number P of hops from the node to the sink node, which is $P = f(n, d_{us}) = n + \lceil \frac{d_{us}}{averd_s} \rceil$, $u \in \{a, b, c, d, e, f\}$. After the above two steps, the routing table and P stored locally by each node are shown in the table in Fig. 3. The farthest node f obtains two 6-hop routing paths to the sink node, but the node d is not found by the neighbor node. We will solve this problem in the third step. The third step continues to run the maximum P-hop route broadcast to construct the initialized reference routing paths. The sink node sends routing broadcast construction information to two neighbor nodes a and b. Each node continues to send routing update information to neighbor nodes when it receives routing update information for the first time or when its own routing table changes. Neighbor nodes will select paths with hops less than or equal to P to join the local storage. The above process continues until no more nodes' routing tables are updated. Finally, the initialization result is shown in step 3 of the table in Fig. 3. At this point, each sensor node obtains all P-hop routing paths to the sink node, and at the same time receives the public parameters broadcasted by the sink node. The entire HWSNs is successfully initialized.

B. Data Encryption and Shares Generation

After initializing parameters and generating routing paths, all nodes in the whole network have established communication channels with the sink node. Then the source node starts to collect data and parallel transfer encrypted shares. This work is divided into two steps. First, the source node randomly selects an invertible cyclic matrix. We first propose a method of generating invertible cycle matrix and verify its feasibility. Then, we describe the process of data encryption and shares generation. The source node encrypts the collected data into ciphertext, using the invertible cycle matrix to split the ciphertext into a group of shares, and randomly selects routes to send messages.

1) Method of Generating an Invertible Cycle Matrix and Its Feasibility: In order to design a lightweight secret sharing scheme, we build an invertible matrix on $\mathbb{F}_2$ and then use the invertible cyclic matrix to construct XOR operations for data encryption and decryption, so as to reduce the computational complexity of the scheme. In this subsection, we will give the generation method of invertible cyclic matrices by using the special properties of cyclic matrices. Then we prove that the cyclic matrices generated by this method are invertible, which provides theoretical support for the lightweight secret sharing scheme as follow subsection.

First, we give a class of cyclic matrices B. Let the first row of matrix B be $b_1 = (b_{1,1}, b_{1,2}, \cdots, b_{1,t})$. Consider the cyclic matrix B:

$$B = \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,t-1} & b_{1,t} \\ b_{1,t} & b_{1,1} & \cdots & b_{1,t-2} & b_{1,t-1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ b_{1,2} & b_{1,3} & \cdots & b_{1,t} & b_{1,1} \end{bmatrix} \tag{11}$$

Let the elements in the first row of the cyclic matrix B be the generators of B, then $f(x) = b_{1,1} + b_{1,2}x + \cdots + b_{1,t}x^{t-1}$ is called the generator polynomial. Then, we guarantee that the cycle matrix B is invertible by choosing a suitable generator polynomial. The theorem and proof used are as follows.

Theorem 1: The necessary and sufficient condition for the invertibility of the cyclic matrix B is $\prod_{i=1}^{t} f(x^i) \neq 0$.

Proof: Consider the Vandermonde matrix:

$$\Lambda = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ x & x^2 & \cdots & x^{t-1} & x^t \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x^{t-1} & (x^2)^{t-1} & \cdots & (x^{t-1})^{t-1} & (x^t)^{t-1} \end{bmatrix} \tag{12}$$

The $\det(\Lambda)$ is not zero, thus Λ is invertible. Consider the diagonal matrix:

$$F = \begin{bmatrix} f(x) & & & \\ & f(x^2) & & \\ & & \ddots & \\ & & & f(x^t) \end{bmatrix} \tag{13}$$

Then we have:

$$B\Lambda = \begin{bmatrix} f(x) & f(x^2) & \cdots & f(x^t) \\ xf(x) & x^2 f(x^2) & \cdots & x^t f(x^t) \\ \vdots & \vdots & \cdots & \vdots \\ x^{t-1} f(x) & (x^2)^{t-1} f(x^2) & \cdots & (x^t)^{t-1} f(x^t) \end{bmatrix} = \Lambda F \tag{14}$$

Consequently, the necessary and sufficient condition for the invertibility of the cyclic matrix B is that F is invertible, i.e. $\prod_{i=1}^{t} f(x^i) \neq 0$. As long as selecting a suitable generator polynomial, the cyclic matrix B is invertible. Finally, to generate the t-dimensional invertible matrix B on $\mathbb{F}_2$, we further

simplify the form of the invertible cyclic matrix B. Below we state the relevant lemma and proof.

Lemma 1: If the sum of the generators of the cyclic matrix B on $\mathbb{F}_2$ is 1, then B is invertible.

Proof: When the sum of the generators of the cyclic matrix B on $\mathbb{F}_2$ is 1, $f(1) = b_{1,1} + b_{1,2} + \cdots + b_{1,t} = 1$, then $\prod_{i=1}^{t} f\left(1^i\right) = 1 \neq 0$ and $\prod_{i=1}^{t} f\left(x^i\right) \neq 0$. Therefore, we generate t-dimensional ($t > 1$) invertible matrix B on $\mathbb{F}_2$ according to the following rules.

- When $t = 2$, $B = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$;
- When $t = 3$, $B = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$;
- When $t \geq 4$, B is the cyclic matrix with $(1,0,1,0,0,\cdots,0,1)$ as the generators. Note that the first, third, and last bits of the generators are 1, and the rest are 0.

In this way, we gain the invertible cycle matrix B on $\mathbb{F}_2$. In the next subsection, using the invertible cycle matrix B given in this subsection, the source node splits the ciphertext into a set of shares. We will describe this process below.

2) Message Encryption and Shares Generation: First, the source node needs to symmetrically encrypt the collected data. In a certain period, the source node w generates the plaintext data dat and runs $\mathcal{SE}.Setup(1^\kappa)$ to generate the random symmetric encryption key key_w, and then conducts $\mathcal{SE}.Enc(dat, key_w)$ to obtain the symmetric encryption ciphertext dat_E. Subsequently, the source node w computes $\mathcal{AE}.Enc(key_w, pk)$ with the sink node's public key pk in PPK. Finally, w constructs the ciphertext:

$$C = \{\mathcal{SE}.Enc\,(dat, key_w)\,, \mathcal{AE}.Enc\,(key_w, pk)\}\,. \quad (15)$$

After that, the source node needs to use a lightweight secret sharing algorithm to split the ciphertext into multiple shares. The source node w sets the number of message blocks as $t(1 \leq t < |T_w|)$ and divides C evenly into t segments $C = c_1 \| c_2 \| \cdots \| c_t$. Each c_i has the same length, i.e., $|c_1| = |c_2| = \cdots = |c_t|$ (If the number of c_t digits is insufficient, zero is padded at the end). w generates the invertible matrix $B = \left(b_1^T, \cdots, b_t^T\right)^T$, where $b_1, \cdots, b_t$ are t row vectors on $\mathbb{F}_2$ according to the method in Section V-B. Take the XOR values of t row vectors for $b_{t+1} = b_1 \oplus b_2 \oplus \cdots \oplus b_t$, then w obtains a vector set of $t+1$ vectors of t-dimensional on $\mathbb{F}_2$, where any t vectors are linearly independent. Note that $b_1, \cdots, b_t$ are linearly independent, so the remaining $t = C_{t+1}^t - 1$ results of t vectors' combinations are all generated by b_{t+1} replacing a vector b_{i^*} in $b_1, \cdots, b_t$. If the replaced $b_1, \cdots, b_{i^*-1}, b_{t+1}, b_{i^*+1}, \cdots, b_t$ are linearly correlation, then because $b_{t+1} = b_1 \oplus b_2 \oplus \cdots \oplus b_t$, so $b_1, \cdots, b_t$ are linearly correlation, which derives a contradiction. After that, w calculates $s_i = b_i[1]c_1 \oplus b_i[2]c_2 \oplus \cdots \oplus b_i[t]c_t$, $(1 \leq i \leq t+1)$ to obtain the shares set $\{s_1, \cdots, s_{t+1}\}$, with a total of $t+1$ elements, and obtaining any t of them can restore the original ciphertext C.

Finally, the source node needs to choose different routes to send the split ciphertext shares. w randomly selects $t+1$ routing paths $path_1, \cdots, path_{t+1}$ from T_w, and constructs the sent message:

$$Meg_i = \{i, s_i, H(i \| s_i), path_i \in T_w,\ path^* = \{ID_w\}, TS(timestamp)\} \quad (16)$$

By this means, the source node completes the work of encrypting and splitting the data collected in one cycle into multiple shares, and then transmits the shares through multiple routing paths.

C. Message Transmission

After the source node sends the ciphertext shares according to the randomly selected routing paths, many relay nodes need to forward the message before the data is transmitted to the sink node. The most important step is how the node selects the appropriate next-hop node among its neighbors. This requires comprehensive consideration of the influence of malicious nodes, reference path, energy load balance and distance from the sink node to find an optimal routing path. At the same time, in order to consider the security of the data, after the relay node forwards the message, a malicious node detection and management mechanism needs to be introduced. When a malicious node is detected, anomaly information will be reported to the sink node and the reputation degree of the node will be updated. Section V-C.1 describes the interaction process and strategy of a node to forward messages to its neighbors in five steps. While Section V-C.2 describes the malicious node detection and management mechanism.

1) Relay Node Selection Method: The way to select relay nodes during message transmission needs to consider a variety of factors, some of which require interaction between nodes for real-time parameter values, such as the nodes' existing energy. We assume that a message Meg_i is transmitted to node u which needs to choose the next relay node to continue the message transmission. Details of the selection process are described below.

① For node u, when trying to send Meg_i to the next-hop, it first sends energy and location query Q_j to all neighbor nodes v_j in the area centered on itself and within the range of D_u. Note $v_j \in Nei_u \backslash Meg_i.path^*, 1 \leq j \leq |Nei_u \backslash Meg_i.path^*|$, where the set of neighbor nodes of u is Nei_u. Nodes that have already forwarded the message Meg_i are not considered. The query Q_j contains three elements ID_u, g^{α_j}, TS, where ID_u is the identity of the node u, $\alpha_j \in_R G_q$ is a random number in G_q, $TS = Meg_i.TS$ is the timestamp of message Meg_i. Finally, u sends $Q_j = \{ID_u, g^{\alpha_j}, TS\}$ to all neighbor nodes in $Nei_u \backslash Meg_i.path^*$.

② After receiving the energy and location query $Q_j = \{ID_u, g^{\alpha_j}, TS\}$ from u, the neighbor node v_j will obtain its own identity ID_{v_j} and the remaining energy $E(v_j)$. And according to the physical location Lo_s of the sink node s and its own physical location Lo_{v_j}, the distance to the sink node $d_{v_j s} = \|Lo_s - Lo_{v_j}\|_2$ is calculated. After that, v_j generates a random number $\beta_j \in_R G_q$ and calculates g^{β_j} as well as $(g^{\alpha_j})^{\beta_j} = g^{\alpha_j \beta_j}$. Finally, v_j calculates the energy and location ciphertext $\mathcal{SE}.Enc\left(ID_{v_j} \left\| E\left(v_j\right) \right\| d_{v_j s} \| TS, g^{\alpha_j \beta_j}\right)$ and returns the encrypted answer A_j to the node u.

$$A_j = \left\{\mathcal{SE}.Enc\left(ID_{v_j} \left\| E\left(v_j\right) \right\| d_{v_j s} \| TS, g^{\alpha_j \beta_j}\right), g^{\beta_j}\right\} \quad (17)$$

③ After node u receives A_j returned by neighbor node v_j, it uses α_j to calculate $\left(g^{\beta_j}\right)^{\alpha_j} = g^{\alpha_j \beta_j}$, and then runs the decryption algorithm $\mathcal{SE}.Dec\left(\mathcal{SE}.Enc\left(ID_{v_j} \left\| E\left(v_j\right) \right\| d_{v_j s} \| TS, g^{\alpha_j \beta_j}\right), g^{\alpha_j \beta_j}\right) = \left(ID_{v_j}, E\left(v_j\right), d_{v_j s}\right)$ to get the current energy of the corresponding neighbor node v_j and the distance $d_{v_j s}$.

After receiving the corresponding responses to all queries, node u obtains a set of energy and distance information $\left\{\left(E\left(v_j\right), d_{v_j s}\right)\right\}$ about neighbor nodes.

④ Thereafter, u computes relay node selection parameter IF for each v_j.

$$IF\left(u, v_j\right) = p_{v_j} \cdot p_a \cdot \frac{E\left(v_j\right)}{E_T\left(d_{uv_j}, L\right)} \cdot \frac{1}{d_{v_j s}^2} \tag{18}$$

Among them, p_{v_j} is the reputation degree of v_j (After the initialization phase, all neighbor nodes are trusted by default, and p_{v_j} is uniformly set to 1. This parameter will be updated after the detection of malicious nodes, as detailed in section V-C. p_a is the reference path selection parameter. If $v_j \in Meg_i.path$ then $p_a = \lambda$, otherwise $p_a = 1$, where $\lambda(\lambda > 1)$ represents the weight coefficient of the reference path selection parameter. This parameter makes u tend to select the next-hop node along the reference path attached to the message Meg_i. After that, u selects the node v_j^* with the largest IF value from the neighbor nodes set $Nei_u \backslash Meg_i.path^*$ as the next-hop relay node.

⑤ Finally, u forwards the message Meg_i' updated according to Meg_i and v_j^*.

$$Meg_i' = \left\{i, s_i, H\left(i \| s_i\right), path_i, path^* \| ID_{v_j^*}, TS\right\} \tag{19}$$

These five steps describe the strategy of the node to forward the message to the relay node. When forwarding the message, the node refers to the reference path attached to the message Meg_i. At the same time, the node calculates the relay node selection parameter IF value by balancing the energy consumption ratio $E(v_j)/E_T(d_{uv_j}, L)$, distance parameter $1/d_{v_j s}^2$, reputation degree p_{v_j} and the reference path selection parameter p_a. Then, the node selects the neighbor node with the largest IF value and forwards the message to it. After that, relay nodes forward the messages in turn. If the sink node receives at least t shares corresponding to the messages, the original ciphertext C can be recovered.

2) Malicious Nodes Management: In Section IV-C Security Model, we describe the possible malicious node attacks. For our LSDT scheme, malicious node v either tampers with the received message, making $Meg' \neq Meg$, or deliberately falsifies its energy value $E'(v)$ so that $E'(v) \gg E(v)$. This will cause its neighbor node u to find that $IF'(u, v) = p_v \cdot p_a \cdot \frac{E'(v)}{E_T(d_{uv}, L)} \cdot \frac{1}{d_{vs}^2} \gg IF(u, v)$. Therefore, u always forwards v as a relay node, forming a message "black hole".

To counteract the attack, we propose an anomaly-based mechanism for detecting malicious nodes. Furthermore, the network updates the reputation degree of each node periodically, enabling the system to choose more trustworthy nodes as the next-hop for message transmission. By implementing this management mechanism, the system can effectively withstand malicious node attacks, as described below.

Malicious Node Detection Mechanism: In WSNs, message propagation is omnidirectional. After the relay node u forwards the message to v_j, v_j also needs to forward the message to the next-hop node, which requires v_j to broadcast the message in the communication range. Therefore, the node u can also receive the message broadcast by v_j. This means that u can monitor the behavior of the next-hop node after forwarding the message. We use this mechanism to design a monitoring and management system for malicious nodes, and the specific mechanisms are as follows.

As shown in Fig. 2, in this section, v is a malicious node, represented by a black hole, and u is the last hop node of v. After u sends Meg_i to v, u will monitor v's behavior. If v does not send the corresponding Meg_i' within the time interval TD, u will mark v as a suspicious node; if v sends the corresponding Meg_i' within the time interval TD, but $H(\{i, s_i\}')$ in Meg_i' is different from $Meg_i.H(i, s_i)$, then u will also mark v as a suspicious node. Finally, u sends $M = \{\text{"}Anomaly\text{"}, ID_v, Lo_v, TS = Meg_i.TS\}$ to the sink node through a multi-hop route that does not pass through v.

Reputation Degree Update Mechanism: ① After receiving the malicious node report M, the sink node will verify the received $Meg_i(Meg_i.TS = M.TS)$. If the report is true, it will recalculate the anomaly value of suspicious node v as $\gamma_v = \gamma_v + 1$ (value γ_v is initially 1). Otherwise, the reporting node is regarded as a suspicious node. ② If the sink node solves t shares of different combinations (this part of work is shown in Section V-D) and finds that one of the received messages Meg_i has an error during a certain transmission through comparison, the sink node marks $Meg_i.path^*$ as a suspicious path. Let l be the number of nodes in the $Meg_i.path^*$, then $\forall v \in Meg_i.path^*, \gamma_v = \gamma_v + \frac{1}{l}$. ③ The reputation degree of each node v is $p_v = \gamma_v^{-k}$, where k is the number of times this node is marked as a suspicious node. ④ Every certain period T_c, the sink node broadcasts the ID and updated reputation degree of the node whose reputation degree has changed to the whole network. Each node u will check its own neighbor node list Nei_u and update the corresponding reputation degree. Due to the avalanche nature of symmetric encryption, if a malicious node tampers with a share, the recovered data will be garbled [39].

When a malicious node appears during a round of data transmission, according to the malicious node detection mechanism, it will be marked as a suspicious node and its reputation degree will be updated. At this time, the IF value corresponding to the malicious node is smaller than that of other neighbor nodes. When the next round of data transmission is performed, the node forwards the message to the neighbor node with the largest IF value, so that it successfully bypasses the malicious node with a smaller IF value.

D. Original Data Recovery

After receiving a certain number of messages, the sink node recovers the complete ciphertext C from shares in these messages by calculating the inverse matrix of the encryption matrix. Then, it obtains the session key by decrypting the $\mathcal{AE}.Enc(key_w, pk)$ in the ciphertext C, and then recovers the original data. The details of this process are described below.

After the sink node receives any t messages $\{Meg_{i_1}, \cdots, Meg_{i_t}\} \subset \{Meg_1, \cdots, Meg_{t+1}\}$, because each message Meg_{i_j} contains the message sequence number $i_j (i_j \in \{1, \cdots, t+1\})$ and the corresponding share s_{i_j}, the sink node gets the t pair values $\left\{\left(i_j, s_{i_j}\right)\right\} \subset \{(i, s_i)\}$.

The sink node also uses the method in Section V-B to generate the corresponding t-dimensional invertible matrix

$B=(b_1^T, b_2^T, \cdots, b_t^T)^T$, and calculates $b_{t+1}=b_1 \oplus b_2 \oplus \cdots \oplus b_t$. Then it gets the vector group $b_1, b_2, \cdots, b_{t+1}$. Note that the sink node and the sensor node use the same algorithm, therefore the generated vector group $b_1, b_2, \cdots, b_{t+1}$ on $\mathbb{F}_2$ are the same, which enables the sink node to obtain the corresponding encryption matrix for each t shares. Take t vectors in $\{b_1, b_2, \cdots, b_{t+1}\}$ corresponding to $\{i_j\}$ to form a t-dimensional invertible square matrix $\Gamma=\left(b_{i_1}^T, b_{i_2}^T, \cdots, b_{i_t}^T\right)^T$ on $\mathbb{F}_2$. The sink node computes the inverse matrix Γ^{-1} of Γ on $\mathbb{F}_2$. After that, the sink node computes on $\mathbb{F}_2$:

$$\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_t \end{bmatrix} = \Gamma^{-1} \begin{bmatrix} s_{i_1} \\ s_{i_2} \\ \vdots \\ s_{i_t} \end{bmatrix} \tag{20}$$

The original ciphertext $C = c_1 \| c_2 \| \cdots \| c_t = \{\mathcal{SE}.Enc(dat, key_w), \mathcal{AE}.Enc(key_w, pk)\}$ is recovered by the above equation (20), where $c_i = \Gamma_i^{-1}[1]s_{i_1} \oplus \cdots \oplus \Gamma_i^{-1}[t]s_{i_t}$. After the sink node solves the session key key_w through the private key sk, it can recover the plaintext data dat.

VI. Security Analysis of LSDT

A complete data transmission system, as security goals stated in Section IV-C, must have high availability, data integrity, and confidentiality at the same time. This section will analyze and prove from a theoretical perspective that the proposed LSDT scheme meets *Definition 3*∼*5*.

A. Data Integrity

In general, LSDT sets up double barriers for data integrity damage:

1) First Layer of Protection: When each share (i, s_i) is sent, the message Meg_i also contains the share's hash value $H(i||s_i)$. The sink node can determine if a message has been tampered with by calculating whether the hash value of $i\|s_i$ contained in Meg_i is the same as $H(i\|s_i)$ contained in Meg_i. We present the data integrity protection claim in this scheme through the *Definition 1* given in Section III-C and *Definition 3* in Section IV-C.

Claim 1: This scheme is data integrity protected when the hash function used is collision resistance.

Proof: During message transmission, Meg_i contains m (where $m = i\|s_i$) and the corresponding hash value $H(m)$. Assuming that adversary $\mathcal{A}$ can destroy the data integrity of the scheme, it indicates that adversary $\mathcal{A}$ has a non-zero advantage $\eta > 0$ to find $m' \neq m$ so that $H(m) = H(m')$. That is, the following formula holds:

$$\Pr\left[H\left(m'\right) = H(m)\right] > \eta, m' \neq m. \tag{21}$$

This is in contradiction with the collision resistance property of the hash function H. Therefore, LSDT scheme utilizes the property of the hash function to ensure data integrity.

2) Second Layer of Protection: Even if the adversary breaks the collision resistance of the chosen hash function so that a share s_i is tampered with, our secret sharing scheme can still recover the original ciphertext data. If the adversary tampers with a message Meg_i, according to the malicious node detection mechanism of our scheme, the upstream node of the malicious node will report to the sink node that the message Meg_i has been tampered with. Then, the sink node uses the remaining t messages $Meg_1, \cdots, Meg_{i-1}, Meg_{i+1}, \cdots, Meg_{t+1}$ to recover the original ciphertext data. If the upstream node of the malicious node does not report the malicious node information, the sink node decrypts the shares set containing the share in tampered message Meg_i to obtain the data dat. However, due to the confusion and diffusion mechanism of symmetric encryption, the final decrypted plaintext data is garbled, and the sink node will find that the message has been tampered with. At this time, the sink node obtains $C_{t+1}^t = t + 1$ possible original data by decrypting any t shares combinations in the received $t + 1$ messages $Meg_1, \cdots, Meg_{t+1}$, and find the one that conforms to the data specification as the original plaintext data.

B. Data Confidentiality

According to the work of Herranz et al. [40], the way encrypting dat to ciphertext C in LSDT is a kind of public key encryption (PKE), which leads to the following lemma.

Lemma 2: If $\mathcal{AE}$ is $NM - CPA$ secure and $\mathcal{SE}$ is $NM - OT$ secure, then PKE is $NM - CPA$ secure, and can also be derived to be $IND - CPA$ secure [40].

Claim 2: Our proposed LSDT satisfies data confidentiality defined in *Definition 4* if the PKE is $IND - CPA$ secure.

Proof: We assume that it occurs within the probability p when the number of elements $t' = |\{(i, s_i)\}|$ in the set $\{(i, s_i)\}$ obtained by $\mathcal{A}$ is less than t. Correspondingly, when the number of elements in the set $\{(i, s_i)\}$ obtained by $\mathcal{A}$ is greater than or equal to t, it occurs within the probability $1 - p$.

Based on these two hypotheses, we first analyze the situation where the adversary obtains different amounts of shares and discuss the probability of the adversary decrypting the original data in the two cases. Finally, the data confidentiality protocol of LSDT is regulated to the $IND - CPA$ secure of PKE through *Lemma 2*.

Case 1: For probability $p < 1$, the number of elements in the set $\{(i, s_i)\}$ obtained by $\mathcal{A}$ is less than t, that is, $t' = |\{(i, s_i)\}| < t$. Assuming that the set of pairs obtained is $\left\{\left(i_j, s_{i_j}\right)\right\}, 1 \leq j \leq t'$, $\mathcal{A}$ can obtain the equations:

$$\begin{cases} b_{i_1}[1]c_1 \oplus b_{i_1}[2]c_2 \oplus \cdots \oplus b_{i_1}[t]c_t = s_{i_1} \\ b_{i_2}[1]c_1 \oplus b_{i_2}[2]c_2 \oplus \cdots \oplus b_{i_2}[t]c_t = s_{i_2} \\ \vdots \\ b_{i_{t'}}[1]c_1 \oplus b_{i_{t'}}[2]c_2 \oplus \cdots \oplus b_{i_{t'}}[t]c_t = s_{i_{t'}} \end{cases} \tag{22}$$

These equations (22) has at least $\left(2^{t-t'}\right)^{|c_i|} = 2^{(t-t')|c_i|}$ possible solutions, so the probability of the adversary choosing the correct solution is $\epsilon_1 = 1/2^{(t-t')|c_i|}$. ϵ_1 is a minimum value, which actually proves that the secret sharing scheme we designed satisfies the *Definition 2* given in Section III-C and is therefore secure.

Case 2: For the probability $1 - p$, the number t' of elements in the set $\{(i, s_i)\}$ obtained by $\mathcal{A}$ is greater than or equal to t, then it can solve the correct ciphertext $C = \{\mathcal{SE}.Enc(dat, key_w), \mathcal{AE}.Enc(key_w, pk)\}$.

Therefore, assuming $Adv[dat \leftarrow \mathcal{A}(C)] = \epsilon_C$, then $Adv\left[dat \leftarrow \mathcal{A}\left(\{(i, s_i)\}\right)\right] \leq \left[p\epsilon_1 + (1-p)\right]\epsilon_C$.

Assuming that the adversary $\mathcal{A}$ has a probability advantage of ϵ_M to obtain the original plaintext data dat from $\{(i, s_i)\}$, then the adversary $\mathcal{A}$ has an advantage of probability that $\epsilon_C = \frac{\epsilon_M}{p\epsilon_1+(1-p)}$ to get the original plaintext data dat from C.

Below we will prove this conclusion by reductio ad absurdum. Suppose that there is a PPT adversary $\mathcal{A}_E$ in PKE. The challenger $\mathcal{C}_E$ generates the public-private key pair pk, sk of the asymmetric encryption $\mathcal{AE}$, and announces the pk. After that, $\mathcal{A}_E$ and $\mathcal{C}_E$ play the following game:

1. $\mathcal{A}_E$ selects two plaintext data dat_0, dat_1 and sends them to $\mathcal{C}_E$.

2. $\mathcal{C}_E$ randomly selects $\mathcal{SE}$'s session key key_w and $b \in_R \{0, 1\}$, then calculates $C = \{\mathcal{SE}.Enc(dat_b, key_w), \mathcal{AE}.Enc(key_w, pk)\}$ and sends it to $\mathcal{A}_E$.

3. $\mathcal{A}_E$ sends C to $\mathcal{A}$, $\mathcal{A}$ returns a dat.

4. $\mathcal{A}_E$ generates a guess $b' \in \{0, 1\}$. If $dat = dat_1$, then $b' = 1$. If $dat = dat_0$, then $b' = 0$. If dat is different from dat_0, dat_1, then $\mathcal{A}_E$ randomly select $b' \in_R \{0, 1\}$.

The probability of $\mathcal{A}_E$ winning the game is $\frac{1+\epsilon_C}{2}$. Thus, $\mathcal{A}_E$ has the advantage $\frac{\epsilon_C}{2} = \frac{\epsilon_M}{2[p\epsilon_1+(1-p)]}$ to win the game. This contradicts the fact that PKE is $IND-CPA$ secure, so *Claim 2* is proved.

C. Data Availability

We propose data availability *Claim 3* of the LSDT scheme corresponding to *Definition 5*.

Claim 3: Take malicious nodes performing black hole attack as an example, our LSDT scheme is capable of rapidly restoring data availability. In particular, for parameter settings as shown in Table I, after introducing malicious nodes, LSDT scheme will restore data availability of the system after up to $7le$ rounds of message transmission. l is the maximum number of hops in a single route and e is the maximum number of malicious neighbor nodes around each relay node.

Proof: At the beginning of the proof, we make the following assumptions:

- The maximum energy of the node that can be contained in the energy and location answer A_j is $Energy_{max}$; The minimum initial energy of the deployed nodes is $Energy_{actual}$; The minimum energy of the healthy nodes is $Energy_{actual}/10$.
- The minimum distance between neighbor sensor nodes is d_{min} and the maximum distance is d_{max}.
- Consider that node u transmits different Meg to its neighbor nodes multiple times.
- The malicious black hole node is located on the path through which Meg are transmitted.

Thus, suppose there is a malicious black hole node among the u's neighbor nodes. In the first round of message transmission, the reputation degree of the neighbor nodes are equal. The IF value of the black hole node v' calculated by u will be at most

$$\max \frac{IF\left(u, v'\right)}{IF(u, v)} = p_a \cdot \frac{10Energy_{\max}}{Energy_{actual}} \cdot \frac{E_T\left(d_{max}, L\right)}{E_T\left(d_{min}, L\right)} \cdot \frac{d_{max}^2}{d_{min}^2}$$

TABLE I
PARAMETER SETTINGS

Parameter	Value
The number of deployment nodes N	100
The size of area $W \times W$	1000 m$\times$1000 m
The location of sink node	$(500, 500)$
Node's initial energy E_0	0.5 J
Energy heterogeneous parameters θ	0.5
E_{elce}	50 nJ/bit
ξ_{fs}	10 pJ/bit/m^2
ξ_{mp}	0.0013 pJ/bit/m^4
D_0	$\sqrt{\xi_{fs}/\xi_{mp}}$

$$= p_a \cdot \frac{10Energy_{\max}}{Energy_{actual}} \cdot \frac{d_{max}^2}{d_{min}^2} \cdot \frac{E_{elce} + \xi_{mp} d_{\max}^4}{E_{elce} + \xi_{fs} d_{min}^2}$$

times that of any other neighbor node v. This ratio is independent of the length of the sent message. Note that, if the malicious node continues to do evil, due to the reputation degree update mechanism described in Section V-C, the anomaly value of the black hole node v' after the message transmission of δ rounds will be $\frac{\gamma_{v'}}{\gamma_v} = \frac{\delta+1}{1}$ times that of the other healthy neighbor node v. The reputation degree ratio is $\frac{p_{v'}}{p_v} = \frac{(\delta+1)^{-\delta}}{1}$. Therefore, in the δth round of data transmission, the IF ratio will be reduced within a critical value, upper bounded by:

$$\frac{IF\left(u, v'\right)}{IF(u, v)} \leq \frac{1}{(\delta+1)^{\delta}} \cdot p_a \cdot \frac{10Energy_{\max}}{Energy_{actual}} \cdot \frac{d_{max}^2}{d_{min}^2} \cdot \frac{E_{elce} + \xi_{mp} d_{\max}^4}{E_{elce} + \xi_{fs} d_{min}^2}$$

When $\max \frac{IF(u,v')}{IF(u,v)} < 1$, u will select a non-black hole neighbor node as the next-hop to forward messages. Might as well set after $\delta = \delta'$ rounds data transmission there holds $\max \frac{IF(u,v')}{IF(u,v)} < 1$. With the longest routing path length l and at most e malicious neighbors around each node in the path, a secure path is established to bypass the malicious node and ensure stable data transmission after at most $el\delta'$ rounds data transmission. If the parameters are set as listed in Table I, it can be calculated that $\delta' = 7$. Therefore, each hop message transmission requires at most $7e$ rounds to evade the black hole node and complete the data transmission normally.

VII. PERFORMANCE EVALUATION OF LSDT

We design a lightweight and secure data transmission scheme against malicious nodes, LSDT, as shown in detail in Section V. To validate that the LSDT scheme meets the design goals stated in Section IV, we conduct extensive theoretical analysis and experiments in this section to comprehensively evaluate the performance of the LSDT scheme. We first describe the experimental environment and parameter settings in Section VII.A. As for the metric of extending network lifetime, we compare and analyze the node computation load in Section VII.B to demonstrate the reduced energy consumption of a single node. Then we test the energy consumption of the entire network in Section VII.B to prove that our scheme achieves load balancing. Finally, we test the change of network package delivery rate (PDR) in Section VII.B, and demonstrate that the LSDT scheme significantly prolongs the

TABLE II
COMPLEXITY ANALYSIS OF SECRET SHARING ALGORITHMS

Alogrithm	Generation	Recovery	Delay of Generation	Delay of Recovery
LSDT	$(t-1)n$XOR	$(t-1)t$XOR	$3n(t-1)T$	$et(t-1)T$
Puneeth *et al.*'s Scheme	$n(2t-3)mul+n(t-1)add$	$t(t-1)mul+t(t-1)div+2(t-1)add$	$6n(7t-10)T$	$48t(t-1)T$
Chen *et al.*'s scheme	$nt\cdot add+t\cdot shift+(n+1)(n-t)/2\ mul$	$O(t^3)(add+mul)$	$(9n^2-3nt+9n-8t)T$	$O(t^3)T$

network lifetime. For the transmission data security, we prove its integrity and confidentiality through theoretical analysis in Section VI, and give the theoretical conclusion of availability, that is, each hop message transmission will require at most $7e$ rounds to eliminate malicious black hole nodes. We will illustrate this conclusion in the experiment of Section VII.C, and compare PDR with similar schemes to verify that LSDT scheme has a better ability to eliminate malicious nodes.

A. Experimental Environment and Parameter Settings

Our experimental platform uses OMNeT++ 6.0, and the experimental device has 16 GB memory and an i7-10710U CPU. In our experiments, 100 nodes are randomly distributed in 1000 m $\times$ 1000 m monitoring area, and the sink node is fixed at the center, while randomly setting the communication radius of each sensor node to be greater than 60 m and less than 140 m. We define the process of all source nodes forwarding the data collected in a cycle to the sink node as a round of data transmission. Then we take each data transmission round as the basic unit of simulation, and all sensor nodes in the network send 4096 bits of data to the sink node in each round. The experimental parameters of our scheme are shown in Table I.

B. Network Lifetime

In order to extend the network lifetime in the LSDT scheme, we propose a lightweight secret sharing algorithm and an energy-balanced routing algorithm with reference paths in Section V, starting from reducing the computing load of a single node and balancing the overall network load. In order to evaluate the single node computing load, we first give the theoretical complexity comparison of these secret sharing algorithms and simulate the efficiency of these algorithms. Then, we prove that our routing algorithm achieves network energy balance by testing the distribution of energy consumption and the rate of surviving nodes throughout the entire network space. Finally, we use PDR as an indicator to quantitatively verify that the network lifetime in our LSDT scheme has been effectively extended.

1) Node Computing Load Reduction: The secret sharing algorithm based on the XOR operation proposed in Section V is more suitable for the underlying hardware circuit logic. The optimization of the IoT architecture significantly reduces the computational burden of the nodes. In order to prove the lightweight advantage of our proposed LSDT scheme secret sharing algorithm, we first make a theoretical comparative analysis of the lightweight secret sharing scheme adopted by LSDT. Puneeth et al.'s [13] and Chen et al.'s [14] schemes are both multi-path schemes based on secret sharing, while Puneeth et al.'s scheme relies on Shamir's secret sharing on $\mathbb{Z}_p$ and Chen et al.'s scheme designs a lightweight

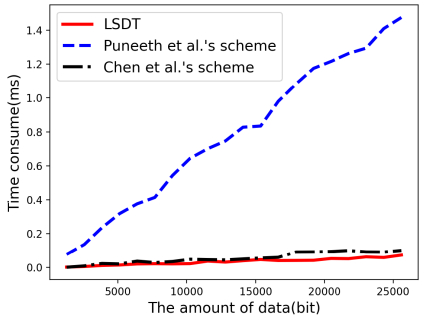


Fig. 4. Time consumption of data shares generation.

algorithm. Compared with these two schemes, all encryption and decryption in our scheme are XOR operations, and this XOR encryption and decryption operation will significantly accelerate the generation of sharing and the recovery of original data.

From the perspective of hardware design, AND gate, OR gate, NOT gate, etc. have one gate delay T from input to output. Shifting 1 bit operation $shift$ also requires a gate delay T. On this basis, we construct NAND gate, Adder and Multiplier [41]. The operation corresponding to the NAND gate is XOR with a delay of $3T$. The Full-Adder corresponds to the calculation of add with the output period $6T$. The Carry-Save Multiplier corresponds to the calculation of mul delay is $18T$, and we regard the Divider div as the same delay as the Multiplier mul. Then we simply calculate the theoretical time of shares generation under the condition of hardware gate circuits, which is shown in Table II. Specifically, under the same node computing capacity, the computation overhead of shares generation by our LSDT scheme is about $1/14$ of Puneeth et al.'s scheme and $1/2$ of Chen et al.'s scheme.

Since the shares generation step is completed by resource-constrained sensor nodes, less computation time represents less resource consumption. In order to further demonstrate the superiority of our LSDT scheme, we conduct a scheme comparison experiment on the time-consuming of shares generation under different data volumes. The simulation results are shown in Fig. 4. The red, blue and black lines represent LSDT, Puneeth et al.'s scheme and Chen et al.'s scheme, respectively. The experiment shows that, with the increase of the amount of data, the shares generation time of the three schemes increases. However, it is worth noting that our lightweight secret sharing scheme takes less time than Puneeth et al.'s scheme and Chen et al.'s scheme in shares generation, and the time-consuming advantage gradually becomes more pronounced with the data volume growing. When the data length increases from 1280 bits to 25000 bits, the average shares generation time of our LSDT scheme is less than $1/14$ of Puneeth et al.'s scheme and $1/2$ of

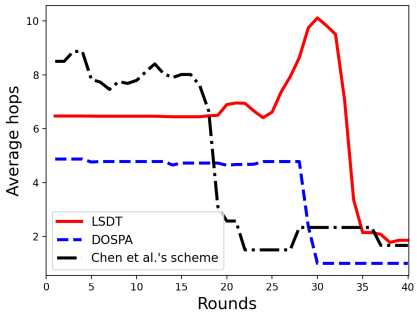
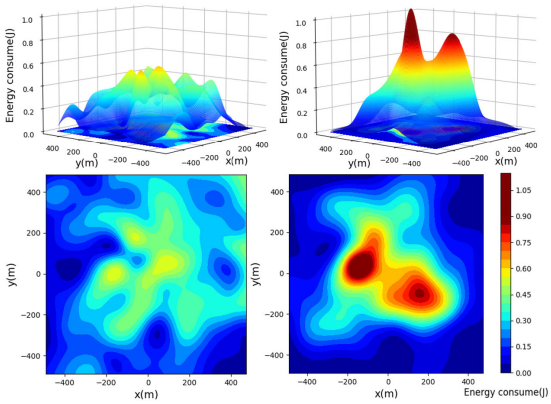


Fig. 5. The average number of routing hops during data transmission.



(a) LSDT (b) DOSPA

Fig. 6. Energy loss distribution diagram of nodes under the same data transmission volume.

Chen et al.'s scheme, which is consistent with the theoretical analysis results. Therefore, experiments show that the LSDT scheme reduces the computational burden of sensor nodes, which significantly reflects the lightweight advantage of our scheme.

2) Balanced Network Energy Consumption: The energy consumption of messages from the source node to the sink node is mainly caused by two factors. One is the internal calculation consumption of all nodes, and the other is the energy consumption of wireless network transmission through the routing path, as energy consumption model shown in Section IV.B. We prove in Section VII.B.1 that our scheme reduces the computational load of the source node, thereby reducing the energy overhead caused by node computing. The wireless transmission energy consumption in the overall HWSNs is positively correlated with the routing length. In theory, the routing length generated by the shortest path algorithm must be the shortest path. Our scheme balances security and energy consumption parameters and may require a longer routing path to deliver messages to the sink node. For this purpose, we experimentally compare the average message routing path hops with Chen et al.'s scheme [14] and DOSPA [42] shortest routing algorithm under the parameter condition of $\lambda = 10$. The results are shown in Fig. 5. Chen et al.'s scheme will first randomly route and forward the message, and then reach the sink node through the dynamic optimal path. DOSPA algorithm will dynamically select the shortest path of data transmission, reducing energy loss and improving transmission efficiency. In the experiment shown in Fig. 5, all nodes of the network in each round send a message to the sink node and calculate the average routing length of each round. That is, each message reaches the sink node after several hops on average. It can be seen that the average routing length of LSDT and DOSPA algorithms remains stable in the first 20 rounds. Chen et al.'s scheme uses some random routing paths. Although the average routing length fluctuates in the first 20 rounds, it is always larger than LSDT and DOSPA.

It is worth noting that after 20 rounds, some nodes in the network run out of energy and cannot perform message forwarding. Since Chen et al.'s scheme and DOSPA scheme do not consider whether the energy reserve of the next node is exhausted, the selected relay node may not continue to transmit, resulting in message truncation. Only messages close to the sink node can be received by the sink node, which reduces the average routing length. However, our energy-load balancing routing algorithm will try to bypass low-energy nodes and successfully send them to the sink nodes, so the average route length goes up slightly. However, in general, our scheme does not significantly increase the average hop count of the route compared to the shortest path routing algorithm, so it does not introduce large energy consumption to the network. Furthermore, since we design the energy load balancing mechanism, the network lifetime of our scheme will be longer than that of Chen et al.'s scheme and the DOSPA scheme, which is demonstrated in subsequent experiments.

In the simulation comparison of Fig. 5, Chen et al.'s scheme and DOSPA scheme show a sharp decrease in the average hop count of messages after 19 and 27 rounds respectively. This shows that these two schemes will quickly consume node energy, a large number of nodes cannot forward messages, and the network lifetime is short. Furthermore, in order to prove that the LSDT scheme designed by us can effectively ensure network load balancing and extend network lifetime, we compare the routing algorithm proposed in this scheme with the DOSPA shortest path routing algorithm for network energy load comparison experiments, as shown in Fig. 6. In this experiment, under the premise that the node energy will not be exhausted, the whole network will transmit 30 rounds of data to the sink node and record the energy consumption of each node. In Fig. 6, X-axis and Y-axis represent the coordinate of each node, while Z-axis represents the energy consumption of nodes. Note that in the comparison experiment, the total amount of data sent by the nodes in the network is the same, that is, the volumes under the surfaces in (a) and (b) are roughly equal, but the highest point of the surface in (b) exceeds 1.0 J, and more than half of the peripheral nodes consume less than 0.2 J, with obvious red areas and large blue areas. This is because the energy loss of the DOSPA [42] in HWSNs is concentrated around the sink node. As the node is far away from the sink node, the energy consumption decreases rapidly, and the edge node has basically no energy loss. The surface in (a) has no prominent peak and the highest value is about 0.4 J. There is no significant fluctuation and fewer dark blue areas. Expect that, the network edge node energy load is small, and most of the remaining nodes' energy loss is more uniform, which illustrates the network load balancing of our scheme. This

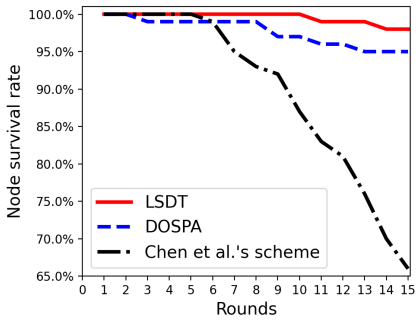


Fig. 7. The node survival rate comparison.

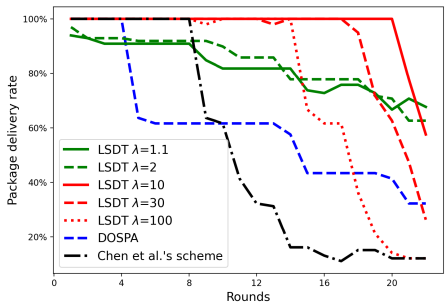


Fig. 8. Network lifetime under different network energy consumption modes.

is due to the fact that the routing algorithm we designed comprehensively balances the "shortest path transmission" and "energy load balancing", ensuring the high availability of the system.

Fig. 6 intuitively shows that the energy consumption of the entire 1000 m × 1000 m HWSNs is more evenly distributed in space under the LSDT scheme proposed by us. It is worth noting that if a node consumes too much energy, it will not be able to continue to perform the message forwarding function, affecting PDR. In order to further demonstrate the advantages of this scheme in balancing network energy consumption and extending network lifetime, we will discuss the survival time of each node when the network continues to perform message transmission. We define nodes with energy more than 20% as survival nodes and define the survival rate of network nodes as the proportion of survival nodes in the total network nodes. Obviously, the more balanced the network energy consumption load, the more the number of survival nodes in the network under the same data transmission round, the higher the survival rate of network nodes. As shown in Fig. 7, we display the proportion of survival nodes in LSDT scheme, DOSPA scheme and Chen et al.'s scheme network under different data transmission rounds. It can be seen that with the increase of data transmission rounds, the network node survival rate of the three schemes is decreasing. However, the LSDT scheme we designed always maintains 100% survival rate of network nodes in the first 10 rounds, while the DOSPA scheme has non-survival nodes in the 3rd round, and Chen et al.'s scheme has non-survival nodes in the 6th round. In general, under the same data transmission rounds, the number of non-survival nodes of the DOSPA scheme is more than twice that of LSDT, while the number of non-survival nodes of Chen et al.'s scheme is nearly ten times that of LSDT. This proves that our LSDT scheme effectively balances the energy load, reduces the occurrence of non-survival nodes, and extends the network lifetime.

3) Network Lifetime Extension: For demonstrating the significant advantages of our scheme in energy load balancing to enhance the lifetime of HWSNs, we conducte comparative experiments on the network lifetime in DOSPA scheme, Chen et al.'s scheme and our LSDT scheme with different weight coefficient of the reference path selection parameter λ, as shown in Fig. 8. We demonstrate the advantage of our scheme in extending the network lifetime by calculating the PDR of the network after each round of data transmission. Overall, as the number of data rounds gradually increases and some nodes in the network run out of energy to complete the forwarding function, the PDR gradually decreases. By observing Fig. 8, it is found that there is no node energy exhaustion in the first 4 rounds of data transmission in DOSPA scheme. From the 5th round, some nodes with large load have insufficient energy and cannot forward messages. The PDR of the network drops rapidly, falling below 50%. Chen et al.'s scheme performs better than DOSPA in the first 8 rounds, but PDR also falls rapidly from the 9th round. The routing algorithm we designed maintains more than 50% PDR before 17th round. In particular, when $\lambda = 10$, our scheme maintains 100% PDR in the first 20 rounds until most of the nodes in the network run out of energy, and the PDR of the network drops below 50% in the 22nd round. It is worth noting that when λ is 1.1 or 2 since the reference path has little effect on the node's choice of the next-hop, the energy and the distance from the next-hop node are mainly considered, resulting in that some messages cannot be transmitted to the area with lower energy near the sink node. Therefore, when λ takes a smaller value, the message transmission effect is not good. When λ is too large, it will be too inclined to choose the shortest path, which is similar to DOSPA shortest path routing. Therefore, when $\lambda = 10$, our scheme effectively balances energy and routing path length, maintains a high PDR, and the network lifetime is more than twice that of DOSPA scheme and Chen et al.'s scheme.

C. Malicious Node Resistance

When there are malicious nodes in the network, package delivery rate (PDR) $\mathfrak{r}$ can effectively reflect the damage to network availability. If the PDR $\mathfrak{r}$ is low during a round of data transmission, it is proved that the data transmission scheme is greatly affected by malicious nodes. Otherwise, the scheme can effectively resist the influence of malicious nodes. When $\mathfrak{r} \leq 50\%$, half of the nodes in the network cannot report data to the sink node, indicating that the attack of malicious nodes has caused the HWSNs to be paralyzed. Our routing path selection algorithm introduces a malicious node resistance mechanism. Our LSDT is compared with Jurado-Lasso et al.'s scheme [43], which does not consider malicious node resistance, and ASA scheme [18], which designs malicious node resistance mechanism, respectively. The number of data rounds and the number of malicious nodes are used as independent variables, respectively, to demonstrate the ability of these scheme to resist malicious node attacks.

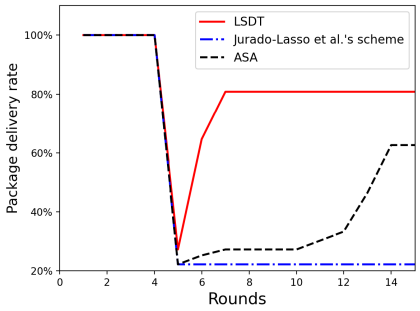


Fig. 9. The effect of resisting malicious node attacks with the same malicious nodes.

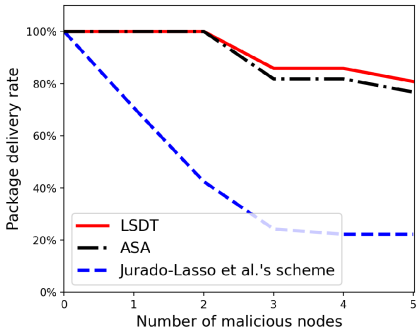


Fig. 10. The effect of resisting malicious node attacks with different number of malicious nodes.

The first is a comparative experiment introducing same malicious nodes, and the results are shown in Fig. 9. The first 4 rounds of data transmission are normal, and the probability of the sink node recovering the original data is 100%. Since then, we introduce five malicious black hole nodes in the 5th round of data transmission, which can attract surrounding nodes to forward messages to themselves and block further transmission of messages. The experimental results declare that the PDR of the network in the 5th round all drops to about 20%, and the systems fall into an unavailable state. Jurado-Lasso et al.'s scheme [43] cannot locate malicious nodes or make subsequent message transmission avoid malicious nodes. Therefore, in the subsequent data transmission rounds, the PDR always maintains at 20%, and the system is completely paralyzed. Since we have designed a malicious node resistance mechanism, the malicious nodes that appear in the 5th round of transmission will be detected and its reputation degree will be reduced. Therefore, starting from the 6th round, the message transmission path in our scheme will choose to bypass the malicious nodes and send the messages to the sink node successfully. In this way, the PDR of our scheme rises rapidly and returns to 80% in the 7th round, and the network turns available. The experimental results are consistent with *Claim 3*. The malicious node detection method based on the posterior probability of ASA also bypasses the malicious nodes, but it takes 8 rounds of data transmission to increase the PDR to more than 50%. This verifies that our LSDT scheme quickly avoids malicious nodes and restores network availability.

As the number of malicious nodes growing, the availability of the system is gradually corrupted. We need to explore the worst-case scenario availability performance, i.e., the PDR performance tendency of the HWSNs system after increasing the number of malicious nodes (which depends on the adversary capability, while the adversary can only corrupt a few nodes in reality). To this end, we compare the PDR of LSDT with Jurado-Lasso et al.'s scheme and ASA scheme after introducing different number of malicious nodes, as shown in Fig. 10. It can be seen that as the number of malicious nodes increases, the PDR of the network shows a downward trend. Specifically, Jurado-Lasso et al.'s scheme exhibits a steep decline. When there are three malicious nodes in the network, the PDR of the network has dropped to nearly 20%. Our LSDT scheme and ASA scheme still maintains a high PDR when introducing malicious nodes, and the PDR of the LSDT scheme is always higher than that of ASA under different number of malicious nodes. Moreover, when the number of malicious nodes increases, the PDR of our proposed LSDT scheme always exceeds 80%. Therefore, the malicious node resistance mechanism in LSDT effectively resists malicious node infringement and ensures the robustness and availability of the network.

VIII. Conclusion

In this paper, the challenges of data transmission in HWSNs are surveyed, which leads to our lightweight and secure data transmission scheme (LSDT) against malicious nodes. First, we design a lightweight secret sharing scheme using XOR operations, which significantly reduces the computational overhead of sensor nodes while improving the robustness of message transmission. Second, we propose a dynamic and efficient malicious nodes detection and management mechanism. This allows routing paths to bypass malicious nodes, avoiding interference. Finally, considering node energy, transmission consumption, and node reputation degree, we design a data transmission scheme based on the reference path, which balances the energy loss of nodes and improves the network lifetime. Security analysis proves that our LSDT scheme effectively protects data CIA security. Theoretical analysis and simulation experimental results verify that our lightweight secret sharing algorithm is more efficient than Chen et al.'s and Puneeth et al.'s schemes, requiring only 1/2 and 1/14 of their computational resources for generating secret shares, respectively. Additionally, our scheme achieves better load balancing and extends network lifetime by more than twice when compared to Chen et al.'s scheme and DOSPA. Furthermore, our multidimensional simulation confirms the effectiveness of LSDT against malicious nodes such as black hole attack, maintaining a packet delivery rate above 80% even under attack.

Moving forward, our future work will focus on extending the LSDT scheme to three-dimensional space, exploring new attack models, and proposing a more cost-effective mechanism for detecting malicious nodes to further enhance the resistance of the network against various attacks.

References

[1] C. Lu et al., "Real-time wireless sensor-actuator networks for industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1013–1024, May 2016.

[2] Z. Lv, B. Hu, and H. Lv, "Infrastructure monitoring and operation for smart cities based on IoT system," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1957–1962, Mar. 2020.

[3] H. Wang, G. Han, Y. Hou, M. Guizani, and Y. Peng, "A multi-channel interference based source location privacy protection scheme in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 2, pp. 2058–2069, Feb. 2022.

[4] O. I. Khalaf, C. A. T. Romero, S. Hassan, and M. T. Iqbal, "Mitigating hotspot issues in heterogeneous wireless sensor networks," *J. Sensors*, vol. 2022, pp. 1–14, Feb. 2022.

[5] P. G. V. Naranjo, M. Shojafar, H. Mostafaei, Z. Pooranian, and E. Baccarelli, "P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks," *J. Supercomput.*, vol. 73, no. 2, pp. 733–755, Feb. 2017.

[6] H. Deghbouch and F. Debbat, "Improved bees algorithm for the deployment of homogeneous and heterogeneous wireless sensor networks," *Int. J. Sensor Netw.*, vol. 38, no. 4, pp. 254–262, 2022.

[7] A. Aziz and K. Singh, "Lightweight security scheme for Internet of Things," *Wireless Pers. Commun.*, vol. 104, no. 2, pp. 577–593, Jan. 2019.

[8] G. Yang and X.-W. Wu, "A lightweight security and energy-efficient clustering protocol for wireless sensor networks," in *Proc. Int. Conf. Ad Hoc Netw.* Cham, Switzerland: Springer, 2018, pp. 237–246.

[9] P. Mishra, N. Kumar, and W. W. Godfrey, "An evolutionary computing-based energy-efficient solution for IoT-enabled software-defined sensor network architecture," *Int. J. Commun. Syst.*, vol. 35, no. 8, May 2022, Art. no. e5111.

[10] X. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li, "Environment-fusion multipath routing protocol for wireless sensor networks," *Inf. Fusion*, vol. 53, pp. 4–19, Jan. 2020.

[11] Z. Alansari, N. B. Anuar, A. Kamsin, and M. R. Belgaum, "A systematic review of routing attacks detection in wireless sensor networks," *PeerJ Comput. Sci.*, vol. 8, Oct. 2022, Art. no. e1135.

[12] Y. Liu, M. Ma, X. Liu, N. N. Xiong, A. Liu, and Y. Zhu, "Design and analysis of probing route to defense sink-hole attacks for Internet of Things security," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 356–372, Jan. 2020.

[13] D. Puneeth, N. Joshi, P. K. Atrey, and M. Kulkarni, "Energy-efficient and reliable data collection in wireless sensor networks," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 26, no. 1, pp. 138–149, 2018.

[14] D. Chen, W. Lu, W. Xing, and N. Wang, "An untraceable data sharing scheme in wireless sensor networks," *Sensors*, vol. 19, no. 1, p. 114, Dec. 2018.

[15] N. Khanna and M. Sachdeva, "A comprehensive taxonomy of schemes to detect and mitigate blackhole attack and its variants in MANETs," *Comput. Sci. Rev.*, vol. 32, pp. 24–44, May 2019.

[16] S. Hamedheidari and R. Rafeh, "A novel agent-based approach to detect sinkhole attacks in wireless sensor networks," *Comput. Secur.*, vol. 37, pp. 1–14, Sep. 2013.

[17] R. K. Sundararajan and U. Arumugam, "Intrusion detection algorithm for mitigating sinkhole attack on LEACH protocol in wireless sensor networks," *J. Sensors*, vol. 2015, pp. 1–12, Feb. 2015.

[18] G. Jahandoust and F. Ghassemi, "An adaptive sinkhole aware algorithm in wireless sensor networks," *Ad Hoc Netw.*, vol. 59, pp. 24–34, May 2017.

[19] M. Radi, B. Dezfouli, K. A. Bakar, and M. Lee, "Multipath routing in wireless sensor networks: Survey and research challenges," *Sensors*, vol. 12, no. 1, pp. 650–685, Jan. 2012.

[20] H.-M. Sun, C.-M. Chen, and Y.-C. Hsiao, "An efficient countermeasure to the selective forwarding attack in wireless sensor networks," in *Proc. TENCON IEEE Region Conf.*, Nov. 2007, pp. 1–4.

[21] M. Sajwan, D. Gosain, and A. K. Sharma, "Hybrid energy-efficient multi-path routing for wireless sensor networks," *Comput. Electr. Eng.*, vol. 67, pp. 96–113, Apr. 2018.

[22] K. Sakthidasan @ Sankaran, X.-Z. Gao, K. R. Devabalaji, and Y. M. Roopa, "Energy based random repeat trust computation approach and reliable fuzzy and heuristic ant colony mechanism for improving QoS in WSN," *Energy Rep.*, vol. 7, pp. 7967–7976, Nov. 2021.

[23] I. Jemili, D. Ghrab, A. Belghith, and M. Mosbah, "Cross-layer adaptive multipath routing for multimedia wireless sensor networks under duty cycle mode," *Ad Hoc Netw.*, vol. 109, Dec. 2020, Art. no. 102292.

[24] W. Lou and Y. Kwon, "H-SPREAD: A hybrid multipath scheme for secure and reliable data collection in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 4, pp. 1320–1330, Jul. 2006.

[25] M. Deryabin et al., "Protocol for secure and reliable data transmission in MANET based on modular arithmetic," in *Proc. Int. Conf. Eng. Telecommun. (EnT)*, Nov. 2019, pp. 1–5.

[26] K. Haseeb, N. Islam, A. Almogren, I. Ud Din, H. N. Almajed, and N. Guizani, "Secret sharing-based energy-aware and multi-hop routing protocol for IoT based WSNs," *IEEE Access*, vol. 7, pp. 79980–79988, 2019.

[27] R. T. Merlin and R. Ravi, "Novel trust based energy aware routing mechanism for mitigation of black hole attacks in MANET," *Wireless Pers. Commun.*, vol. 104, no. 4, pp. 1599–1636, Feb. 2019.

[28] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha, "Light-weight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 62, pp. 112–127, Feb. 2016.

[29] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.

[30] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," *IEEE Trans. Mobile Comput.*, vol. 9, no. 7, pp. 941–954, Jul. 2010.

[31] A. Liu, Z. Zheng, C. Zhang, Z. Chen, and X. Shen, "Secure and energy-efficient disjoint multipath routing for WSNs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 7, pp. 3255–3265, Sep. 2012.

[32] Y. Wang, X. Li, X. Zhang, X. Liu, and J. Weng, "ARPLR: An all-round and highly privacy-preserving location-based routing scheme for VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16558–16575, Sep. 2022.

[33] X. Li, J. Niu, M. Z. A. Bhuiyan, F. Wu, M. Karuppiah, and S. Kumari, "A robust ECC-based provable secure authentication protocol with privacy preserving for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3599–3609, Aug. 2018.

[34] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[35] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[36] G. Chinnaraju and S. Nithyanandam, "Grey hole attack detection and prevention methods in wireless sensor networks," *Comput. Syst. Sci. Eng.*, vol. 42, no. 1, pp. 373–386, 2022.

[37] A. J. C. Sunder and A. Shanmugam, "Jensen–Shannon divergence based independent component analysis to detect and prevent black hole attacks in healthcare WSN," *Wireless Pers. Commun.*, vol. 107, no. 4, pp. 1607–1623, Aug. 2019.

[38] S. Samonas and D. Coss, "The CIA strikes back: Redefining confidentiality, integrity and availability in security," *J. Inf. Syst. Secur.*, vol. 10, no. 3, pp. 1–25, 2014.

[39] C.-H. Tsai and P.-C. Su, "Multi-document threshold signcryption scheme," *Secur. Commun. Netw.*, vol. 8, no. 13, pp. 2244–2256, Sep. 2015, doi: 10.1002/sec.1169.

[40] J. Herranz, D. Hofheinz, and E. Kiltz, "Some (in) sufficient conditions for secure hybrid encryption," *Inf. Comput.*, vol. 208, no. 11, pp. 1243–1257, 2010.

[41] M. J. Bellido-Diaz, J. Juan-Chico, A. J. Acosta, M. Valencia, and J. L. Huertas, "Logical modelling of delay degradation effect in static CMOS gates," *IEE Proc. Circuits, Devices Syst.*, vol. 147, no. 2, pp. 107–117, Apr. 2000.

[42] A. A. Jovith, S. V. K. Raja, and A. R. Sulthana, "Interference mitigation and optimal hop distance measurement in distributed homogenous nodes over wireless sensor network," *Peer Peer Netw. Appl.*, vol. 13, no. 4, pp. 1109–1119, Jul. 2020.

[43] F. F. Jurado-Lasso, K. Clarke, A. N. Cadavid, and A. Nirmalathas, "Energy-aware routing for software-defined multihop wireless sensor networks," *IEEE Sensors J.*, vol. 21, no. 8, pp. 10174–10182, Apr. 2021.