A Key Management Protocol for Heterogeneous Sensor Networks Based on Zero Trust Security and Chaotic Neural Networks

Guogang Li
Huaqiao University
Tong Xie (xthqu98@163.com)
Huaqiao University
Cheng Zou
Huaqiao University
Wenlong Fu
Huaqiao University

Research Article

Keywords: Heterogeneous sensor networks, Blockchain, Zero-knowledge proof, Zero trust security, Chaotic encryption scheme, Security levels

Posted Date: October 18th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2168733/v1

A Key Management Protocol for Heterogeneous Sensor Networks Based on Zero Trust Security and Chaotic Neural Networks

Guogang Li[1,2], Tong Xie[1,2*], Cheng Zou[1,2†] and Wenlong Fu[1,2†]

[1*]Institute of Information Science and Engineering, Huaqiao University, Xiamen, 361021, China.
[2]Xiamen Key Laboratory of Specialized Integrated Circuit Systems, Xiamen, 361021,China.

*Corresponding author(s). E-mail(s): xthqu98@163.com;
Contributing authors: lgg@hqu.edu.cn; zc_onepiece@sina.com; fwl@stu.hqu.edu;
†These authors contributed equally to this work.

Abstract

Aiming at the node security risks and key management vulnerabilities in heterogeneous sensor networks, a key management protocol for heterogeneous sensor networks based on zero-trust security and chaotic neural networks (KMPHSN-ZTSCNN) was proposed. Based on the singular matrix decomposition of difficulty and Hopfield overload chaos neural network classification features, using blockchain and zero-knowledge proof to realize sensor network node registration and authentication, it relies on channel state information (CSI) and adjustable mathematical function to generate dynamically changing keys to complete continuous verification and achieve zero-trust security authentication to ensure data security. The protocol can dynamically allocate different keyspace sizes according to the security level of the group, node storage capacity and computing capacity, and can adapt to the asymmetric structure of heterogeneous sensor networks. Theoretical proof and experimental performance analysis show that the protocol is feasible and can meet the security requirements of heterogeneous sensor networks.

Keywords: Heterogeneous sensor networks, Blockchain, Zero-knowledge proof, Zero trust security, Chaotic encryption scheme, Security levels

1 Introduction

Wireless sensor network (WSN), consisting of multiple sensor wireless terminals with scattered spaces, is one of the core technologies used in the Internet of Things (IoT) by allowing the collection of environmental and physical conditions and coordination of the network through wireless modes. Most of the early WSN research considered homogeneous sensor networks, where all sensor nodes have the same capabilities in terms of communication, computation, memory storage, energy supply, reliability, etc. However, the performance and scalability of homogeneous self-organising networks are poor[1]. As a result, later deployed WSN systems have followed more of a heterogeneous design, mixing together sensors with widely different capabilities[2].

The current heterogeneous sensor network (HSN) applications are at risk due to the lack of a unified and effective security mechanism and security architecture [3], and how to effectively manage the keys in HSNs with different nodes having different battery capacity, communication bandwidth, storage space and computing power is a current challenge. On the other hand, sensor nodes have limited energy, data processing, storage and communication capabilities, HSN use wireless methods to transmit information, and most sensor nodes are deployed in unattended areas lacking supervision, and random access to sensor nodes is also prone to a series of security problems, so traditional security protection mechanisms cannot be fully applied to HSNs [4]. Zero-trust security as a new network security technology framework follows the principle of "never trust, always verify", and uses identity as the basis for access control, which can be established when a node first joins the network with neighboring nodes, backbone nodes and other trust and authentication mechanisms. In addition, it can be combined with blockchain distributed storage technology to provide digital identity for zero trust, establish tracking and recording of node identity information and access behavior, and audit abnormal and attack behavior mechanisms. Combining the above discussion, we propose a key management protocol for heterogeneous sensor networks based on zero-trust security and chaotic neural networks. The protocol consists of five stages: node registration and authentication, security level determination of the sensor network group, establishment of a shared key, continuous verification and node exit. The protocol sets up registration and authentication before a node joins the sensor network, followed by security level determination for the sensor group, and provides criteria for the subsequent space size for shared key establishment. continuous verification is performed on nodes throughout their lifecycle in the sensing network to achieve zero-trust security.

In this research, we have made the following contributions.

1.Our proposed key management protocol combined with the three-point estimation method proposes to use a non-deterministic information evaluation method to determine the security level of a HSN groups, and dynamically assign different keyspace sizes according to the security level of the group, the storage capacity of the nodes and the computing capability, and use a public-key-based asymmetric encryption protocol for data encryption on resource-limited nodes, partially implemented using a cloud server chaotic encryption scheme for neural networks, which avoids the repeated computation of chaotic attractors of sensor nodes in encrypted communication networks, thus greatly saving the energy and resource consumption of sensor nodes.

2.The traditional boundary-based security architecture cannot effectively meet the growing security requirements of today's IoT, and the sensing network has limited storage and processing capacity of sensor nodes. To meet the security requirements, lightweight authentication is required, and we propose a CSI-based periodic key update to provide dynamic functionality to ensure continuous verification and thus achieve the security of the sensing network in a zero-trust environment.

3. We introduce blockchain technology into HSN to register nodes before node authentication and manage node identity information to achieve tamper-proof and traceable identity and access behavior information. We apply zero-knowledge proof to blockchain to achieve node security authentication, privacy protection and node anonymity to improve the security level. In addition, we use blockchain technology to provide digital identity information for achieving zero-trust security, enabling fine-grained control of access, simplifying security algorithms and reducing the computational load of continuous verification.

The remainder of this paper is organized as follows: we discuss related work in Section 2. In Section 3, we present the system model, the threat model and the design goals. The design and workings of the proposed key management protocol are presented in Section 4, followed by the security analysis and performance evaluation in Section 5. Finally, our work is summarised in Section 6.

2 Related work

In the past decades, advances in WSN group key management have focused on the centralised and distributed management of encryption/decryption keys [5]. In the literature, few studies have proposed key management solutions based on zero-trust and neural network chaotic encryption to improve one or more aspects of the HSN security domain.We will introduce some related work from two aspects of key management in sensor network and zero trust security.

2.1 Key management in sensor networks

Key management is the core mechanism for securing WSN services and applications. Key management can be defined as a set of processes and

mechanisms that support key establishment and maintenance of ongoing key relationships between valid parties according to a security policy [6].

Initially, key pre-distribution and key pre-installation were proposed to solve the key management problem in WSNs, where pairs of keys could be established between sensors using pre-distributed keys. The certificate-free public key cryptography (CL-PKC) proposed by Al-Riyami et al [7] eliminates the requirement of name implied certificates. It avoids the administrative cost of certificate distribution and verification and is more suitable for WSNs consisting of low performance sensors. Eltoweissy et al [8] propose an EBS-based dynamic key management scheme that does not use any location information when generating new keys. Once a node is captured, other nodes in the same cluster will perform key updates through a local key update mechanism, thus preventing the compromised node from communicating with it. Zhang et al [9] propose a series of pre-distributed and locally collaborative group key update schemes based on PCGR to address the node leakage problem. Two cascading schemes are also proposed, where each node needs to store the secret shared polynomial of its immediate neighbor nodes, but with high consumption and no guarantee of forwarding secrecy. Divya et al [10] propose a dynamic key management approach based on Hamming distance to address the problem that capturing a few nodes can reveal most of the keys, relying on a centralized key generation gateway to perform key updates, but the key generation gateway's capture provides more sensor node keys to the adversary than the capture of conventional sensor nodes. Schemes [11] and [12] used identity-based cryptosystems to eliminate certificate management overheads. However, they require expensive bilinear pairing operations. In addition, the direct derivation of public keys in ID-PKC leads to the need for key escrow for users and brings the possibility of catastrophic consequences from PKG master key leakage. Seo et al [13] propose a key management scheme for clustered WSNs based on scheme [12] and their previous work. The scheme achieves efficient node authentication, pairwise key establishment between nodes and cluster key update. Tian et al [14] improve the key management scheme proposed in the literature [13] using blockchain technology. They use a blockchain to record information about registered nodes, thus improving the security of the system. However, the solution suffers from a single point of failure problem as it relies on a centralised key generation centre. This problem also leads to poor scalability and is not suitable for large-scale WSNs. The literature [15] proposes an efficient and secure hierarchical decentralized key management (HiDeKM) scheme based on blockchain technology. HiDeKM is efficient and scalable due to its hierarchical structure, which helps to achieve scalability in key management for wireless sensor networks. It allows multiple ESs to act as cluster heads to perform key management in a decentralized manner. The literature [5] proposes an algorithmic solution for group key management in WSNs to address single points of failure in network security, introducing a family key paradigm and end time tickets as a context to operate and execute algorithmic decisions in key management.

2.2 Zero trust security

The term "zero trust" was formally coined by Kindervag in 2010 [16]. Zero trust provides a set of concepts and ideas that reduce the uncertainty of the accuracy of decisions made when executing each access request in information systems and services, assuming that the network environment has been compromised. The Zero Trust Architecture (ZTA) is a plan for enterprise cyber security that is based on the Zero Trust concept and is built around its component relationships, workflow planning and access policies. A model designed on the basis of the Zero Trust Architecture has several advantages. Firstly, the use of multiple factors in authentication allows for better protection of resources and data from attacks and corruption [17]. Secondly, the fine-grained segmentation of access prevents the movement of attackers and malware through the network [18]. Thirdly, resources can also be better protected against DDoS attacks. External and unauthorised requests can be immediately denied as access is only available via zero-trust PEP [19]. Fourthly, improved authentication mechanisms and clear definitions of access policies allow for precise control of access to the network. This allows for a more granular access and permission model than existing solutions [20]. Fifth, by continuously logging and monitoring traffic, suspicious behaviour and attacks can be detected and interrupted faster. At the same time, zero trust increases the traceability of forensics, thus allowing learning from past events [16].

Zero trust may be particularly important for the security of IoT applications, as it often involves networks where an increasing number of devices require flexible security concepts [17]. Zero trust is even considered to be a prerequisite for the "Internet of Everything" [21]. Mehraj et al [18] propose an approach to zero-trust based authentication systems. In addition, Albuali et al [22] proposed a zero-trust identity management model that includes behavioural analysis and multi-factor authentication to determine the trustworthiness of nodes. Sateesh et al [23] propose an SDP architecture for vehicular self-organizing networks to enable the spontaneous creation of networks between vehicles. Several researchers implemented proofs of concept to demonstrate the applicability of zero-trust to the IoT [24–27]. Chen et al [28] propose a blockchain-based zero-trust security protection scheme for the power IoT, enabling strict management of control information, more standardized management of access rights to data, and much higher trustworthiness and interaction rates of business data. In addition, smart home applications are seen to benefit from zero trust as different devices on the network are used to communicate with cloud services. To ensure that access to services from unauthorized nodes in the network is blocked, the user's smartphone can act as an SDP controller, mediating communication between the IoT and the cloud [29]. To overcome the challenges associated with centralized authentication instances, Samaniego et al [30] propose blockchain-based middleware that handles access authentication in a decentralized manner. In addition, the zero-trust principle can also be used to transfer sensitive data. sultana et al [31]

develop a framework that uses zero-trust and blockchain technology to enable the secure exchange of sensitive medical images.

3 Problem statement

In this section we will define the system model, threat model and design objectives for our proposed solution.

3.1 System model

The heterogeneity of the HSN is mainly manifested in the following aspects: computing power, storage capacity and node energy, communication capability and communication protocols, security requirements and external environment. The HSN uses a hierarchical cluster topology as shown in 1 Fig. 1. In terms of the logical structure of the network, the HSN is divided into a total of three layers, namely the base station layer, the cluster head layer and the sensing layer. In terms of the physical architecture, there are many groups in the network, each containing a cluster head node and n sensor nodes. The sensor nodes in the sensing layer act as collectors or monitors of object data information, the cluster head layer nodes act as aggregators and forwarders of object data information, and finally, each cluster head node in the cluster head layer transmits the object data information to be collected or monitored in its group to the base station in the base station layer.

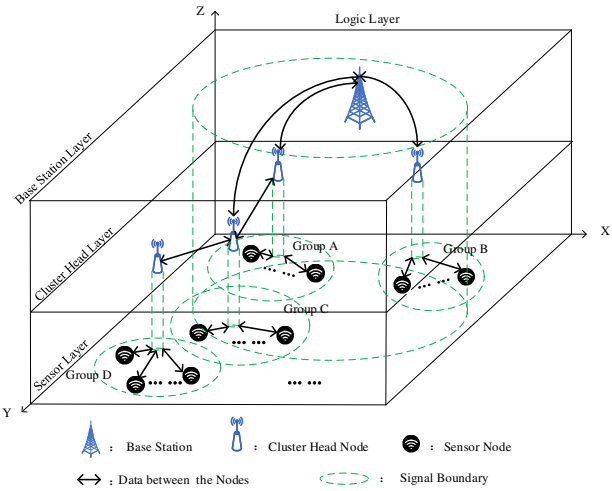

Fig. 1 General model of a heterogeneous sensor network

The system model of KMPHSN-ZTSCNN consists of sensor nodes, cluster head nodes, base stations, cloud servers and blockchains, as shown in Fig. 2.

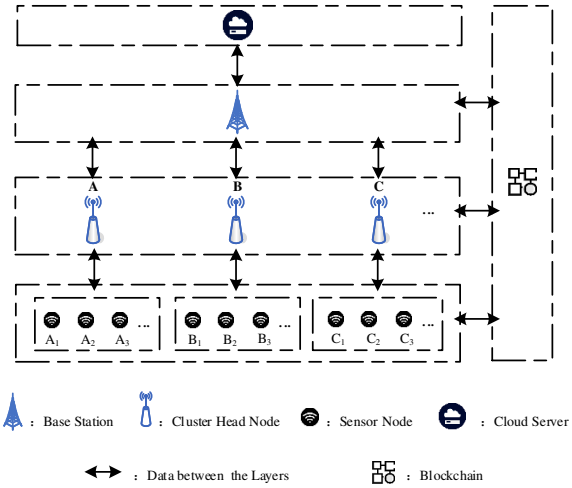

Fig. 2 System model of KMPHSN-ZTSCNN

Sensor node: acts as a collector or monitor of object data information.

Cluster head node: acts as the aggregator and forwarder of object data information and performs continuous identity verification of the sensor node.

Base station: The cluster head node transmits the object data information to be collected or monitored within its group to the base station, where the base station server acts as the CA authorization center and performs continuous verification of the cluster head node.

Cloud server: Chaos attractor outsourcing cloud computing needs to be performed at the cloud server during shared key establishment to reduce the computational load.

Blockchain: used as data storage for the system, verifies node identity through smart contracts, is tamper-proof, and provides traceability of node behavior.

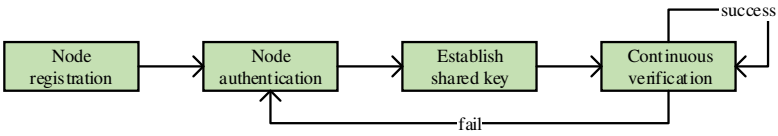

Fig. 3 Life cycle of sensor nodes and cluster head nodes

Node registration and authentication are required before the sensor node and cluster head node enter the network, and the node identity information will be processed for up-chaining. Subsequently, a shared key is established between the sensor node and the cluster head node as well as the cluster head node and the base station, and the node enters the continuous verification process. Once the continuous verification is abnormal, the node will undergo a new authentication process to ensure that each node has zero trust throughout its life cycle, as shown in Fig. 3.

3.2 Threat model

HSNs are usually deployed in relatively low-security scenarios, so sensing information is exposed to a variety of threats such as tampering and discarding. Based on the OSI model, the threats faced by HSNs are analysed to include:

(1) Security threats at the network routing layer

Attackers input a large amount of deceptive and false routing information into the sensing network by intercepting and tampering with the routing information of HSN nodes. Or they disguise themselves as a legitimate aggregation node in the network and send routing requests to neighbouring nodes, causing confusion in the transmission of some messages in the sensor network, repeatedly receiving duplicate routing information, resulting in loop line routing, increased network latency and node energy imbalance.

(2) Security threats at the data link layer

The main way of security threat at the link layer is packet corruption, where an attacker generates a failed ACK by corrupting a byte in the packet, making it impossible for the receiver to properly verify it. and for most MAC protocols, a failed ACK will result in an exponentially growing number of retransmissions, thus reducing the transmission efficiency of the MAC.

(3) Security threats at the physical layer

As HSNs are increasingly used in a wide range of applications, they are deployed in very different environments and are often deployed in unattended environments that are vulnerable to physical capture. Because of the large volume and low cost of sensor node deployment, once an attacker obtains a sensor node, it is easy to obtain sensing information within the network and may also place a new malicious node to replace the original one, posing a huge threat to network security.

3.3 Design objectives

Information security is a fundamental requirement to ensure the privacy of HSNs, and overly complex security mechanisms are difficult to apply to HSNs due to the limitations of the nodes' own computing power, storage space, communication capabilities and energy storage. Like traditional sensing networks, HSN security must meet the network security requirements of confidentiality, availability, integrity, authentication and capture resistance. The main design goals of the KMPHSN-ZTSCNN are as follows.

-Privacy: Data transmission between nodes is encrypted by the established shared key. The privacy of nodes and data should be ensured, and the data transmitted by nodes requires that only legitimate nodes can understand the information received, while illegal nodes cannot understand the information contained in the data even if they obtain it.

-Freshness: The data transmitted by the sender to the receiver should be the latest data generated within the most recent time. Freshness is also reflected in the fact that the keys shared by both communicating parties during the key establishment process are up-to-date.

-Authentication: The identity of the nodes uploading (accessing) the blockchain should be verified to prevent illegal nodes from uploading (accessing) the blockchain data.

-Access control: The attributes of the nodes accessing the blockchain should be verified to be eligible to prevent unauthorised nodes from accessing system data.

-Key management: A secure communication network should meet the need for dynamic update of keys, including operations such as generation, negotiation, storage, distribution, update and revocation of node keys.

-Integrity and auditability: The system should ensure the integrity and auditability of system data to ensure that it is not easily tampered with and can be easily audited.

-Traceability: It should be possible to trace the origin of node data and, if necessary, to locate the location of malicious nodes and deal with them in a timely manner.

-Efficiency: KMPHSN-ZTSCNN aims to provide efficiency in terms of.

(i) computational cost, i.e. the use of lightweight processes to upload and access evidence in the blockchain.

(ii) communication overhead, i.e. the total length of blockchain data transactions should be as short as possible to save network bandwidth.

4 Proposed programme

4.1 Design objectives

Since the security level classification of groups in sensor networks is inherently ambiguous, and there are no quantitative criteria for heterogeneity elements such as the computational power, storage capacity and security sensitivity of nodes, this paper proposes the use of a non-deterministic information evaluation method to determine the security level of heterogeneous sensor network groups in combination with the three-point estimation method.

The subject of the judgment is the HSN group. Several factors of the adjudication metrics are defined, such as the following (four as an example).

Factor 1: The computational capacity of the nodes of the sensing layer group.

Factor 2: Storage capacity of the nodes in the sensing layer group.

Factor 3: Security requirements of the perception layer group.

Factor 4: The environment in which the nodes of the perception layer group are located.

Judgment principles include:

1.The higher the computational capacity of the nodes in the group, the higher the complexity of the cryptographic operations that can be withstood, and the higher the security level that can be judged.

2.The larger the storage capacity of the nodes in the group, the more keys can be stored and the larger the key space can be, the higher the security level can be judged.

3.The higher the security requirements of the group, the higher the security level is determined.

4.The higher the confidentiality requirements of the environment in which the nodes in the group are located, the higher the security level is judged.

Group security level determination method:

1.Determine the index factors: $I_i(i = 1, 2, 3, 4)$,1.where I_i is the four determination index factors mentioned above.

2.Decide the determination grade domain: $L = \{L_1, L_2, L_1, L_4\}$

3.Adopt the three-point estimation method [32] to determine the weight vector of factors.

$W = (w_1, w_2, w_3, w_4)$ is the expression for the evaluation element, the element w_i in W is essentially the degree of affiliation of the element I_i to the group being evaluated. The relative order of importance among the evaluation elements is determined by the three-point estimation method, from which the weight coefficients can be determined and normalised to give $\sum_x^+ w_i = 1\,(w_i \geq 0, i = 1, 2, 3, 4)$

The steps for determining the weight vector using the three-point estimation method are as follows:

1.Treating the weights of the determination elements as approximately normally distributed random variables.

2.Obtain the weight of each judgment indicator element based on the assessor's score and derive the average value m of the judgment indicator elements.

3.Averaging the mean a and b from the sequence of weights less than and greater than m, respectively.

4. m is twice as likely as a , then the average of m and a is $(a+2m)/3$ m is twice as likely as b, then the average of m and b is $(b+2m)/3$. 5.The average of the two points $\bar{x} = (a + 4m + b)/6$.

Taking the estimate value $\bar{x}_i$, which is derived according to the three points of a, m and b, as the estimate value of the factor weight and carrying through the normalization processing, the weight allocation of the factor indexes can be derived.

The relationship matrix is obtained by the assessor quantifying the group being adjudicated at each security level by scoring the degree of affiliation between the individual adjudication indicator elements of the group and the

security level hierarchy domain, and is expressed as:

$$R = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix}_{ij} \tag{1}$$

The elements in row i and column j of the relationship matrix R represent the affiliation of a determined group from a single determination indicator element I_i to a subset of the security level L_i is proposed to be assessed using the assessor interval scoring method, where the point estimate of the element is determined from the assessed interval [33] . The assessment of i judgment indicator elements of j security level subsets by N assessors results in N assessment intervals: $[A_{ni}, B_{ni}]_j$, and the mean point estimate for each sequence of assessment intervals is:

$$E_{ij} = \frac{\sum_{n=1}^{N} \left[{B_{ni}}^2 - {A_{ni}}^2\right]}{2\sum_{n=1}^{N} \left[B_{ni} - A_{ni}\right]} \tag{2}$$

Further, the estimated values of the judgment indicator elements for each assessment interval are derived and normalised to obtain $x_{ij} = \frac{E_{ij}}{100}$.

Using the decision model $M(\cdot, +)$ [32], the weight vector W of the decision indicator elements is synthesised with the relationship matrix R to obtain a comprehensive decision result vector for the safety level of the group being decided: $A = W \cdot R$. Finally, the safety level of the group being decided can be decided according to the principle of maximum subordination.

4.2 Node Registration and Authentication

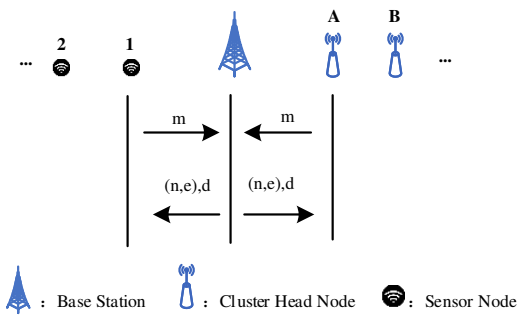

Fig. 4 Node registration process

Before a node can join the blockchain network, it needs to register as a node, and we choose the base station server as the CA authorization centre. The specific steps of registration are as follows: (i) The node sends its own physical information m , which cannot be forged, to the base station as CA; (ii) CA randomly selects two large prime numbers p and q, calculates $n = p \times q$, and forms the public key (n, e) from n and a random integer e. And the value of d is obtained from $ed = 1 \bmod \varphi(n)$ as the private key; (iii) the digital signature of CA on is obtained from RSA encryption algorithm as $s = m^d (\bmod n)$; (iv) CA sends the public and private keys and the ID certificate to the node, that is, the registration is completed. The process is shown in Fig. 4.

The zero-knowledge proof based on RSA is used in the authentication phase from the sensor node to the cluster head node to achieve anonymity of the sensor node to the verifier cluster head node. The sensor node performs the following interactive zero-knowledge proof protocol:

$$ZPK\left\{\alpha \mid \alpha^e \equiv H(m) \bmod n\right\}\left(P_{id} \| \text{ Timestamp } \| \text{ Nonce }\right) = \{c, k\} \in \{0,1\} \times Zn^* \tag{3}$$

Where α represents the secret information of the sensor node, which is the RSA signature of the identity information m, $H(\cdot)$ is the public collision-resistant hash function of $\{0,1\}^* \rightarrow \{0,1\}^l$, n, e and $H(m)$ are shared information,P_{id} represents the identity of the sensor node, which can be a fixed IP or a name marked in a public key certificate, etc.,$Timestamp$ is the timestamp marking the zero-knowledge proof, and $Nonce$ is a one-time random number to prevent replay attacks. The specific certification process is as follow:

(1)The cluster head node optionally $r_1 \in_R Z_n^*$ sends it to the sensor node.

(2)sensor node optional $r_2 \in_R Z_n^*$, calculation: $s_1 = r_1^d \bmod n$.

$$c = H\left(m \| n \| e \| P_{id} \| \text{ Timestamp } \| \text{ Nonce } \| r_1 \bmod n \| r_2^e \bmod n\right) \tag{4}$$

And use their own digital signature s to calculate $k = \frac{r_2}{s_1} \bmod n$, then $\{s_1, c, k, P_{id}$, Timestamp, Nonce $\}$.

(3)The sensor node passes $\{s_1, c, k, P_{id}$, Timestamp, Nonce $\}$ to the cluster head node.

(4)After receiving the zero-knowledge proof evidence, the cluster head node verifies whether the following equation is true:

$$c = H\left(m \| n \| n \| P_{id} \| \text{ Timestamp } \| \text{ Nonce } \| s_1^e \bmod n \| k^e H^c(m) \bmod n\right) \tag{5}$$

If the equation holds, the cluster head node believes that the sensor node has the digital signature and accepts the proof. After several times of zero-knowledge proof, if the verification is successful, it indicates that the identity of the sensor node is true and legal and can join the blockchain network.

4.3 Creating a shared key

4.3.1 Establishment of shared keys between the base station layer and the cluster head layer

At deployment time, each cluster head node is initialized with its own unique private key H_n. The base station initialises the concatenation synapse matrix T, the private key H and the private keys of all cluster head nodes H_n and exposes the concatenation synapse matrix T. Then, the shared key $\hat{T}_{total}$ and the key exchange concatenation matrix for each cluster head node are calculated as follows:

$$\begin{aligned}
&\hat{T}_{total} = H_a * H_b * H_c * \ldots * H_n * H * T * H' * H'_n * \ldots * H'_c * H'_b * H'_a \\
&\hat{T}_a = H_b * H_c * \ldots * H_n * H * T * H' * H'_n * \ldots * H'_a * H'_b \\
&\hat{T}_b = H_a * H_c * \ldots * H_n * H * T * H' * H'_n * \ldots * H'_c * H'_a \\
&\hat{T}_c = H_a * H_b * \ldots * H_n * H * T * H' * H'_n * \ldots * H'_b * H'_a \\
&\ldots \\
&\hat{T}_n = H_a * H_b * \ldots * H_{n-1} * H * T * H' * H^*_{n-1} * \ldots * H'_b * H'_a
\end{aligned} \tag{6}$$

Where $\hat{T}_a, \hat{T}_b, \hat{T}_c, \ldots, \hat{T}_n$ are the link synapse matrix of key exchange of cluster head node $a, b, c, \ldots$ respectively.

The base station encrypts $\hat{T}_{total}$ and transmits it to the cloud server, which substitutes $\hat{T}_{total}$ into the chaotic neural network model:

$$S_i(t+1) = \delta\left[\sum_{j=0}^{N-1} T_{ij} S_j(t) + \theta_j\right], i = 0, 1, 2, \cdots, N-1. \tag{7}$$

$S_i(0) = (0, 0, \ldots, 0)_{l \times n}$ is used as the initial value of the neural network neuron, and the output of the previous state is used as the input of the next state. The cycle is iteratively calculated until the output no longer changes, i.e.$S_i(t+1) = S_i(t)$, and a chaotic attractor is obtained. The next chaotic attractor will be calculated as the new initial value. The $2n$ attractors corresponding to $\hat{T}_{total}$ calculated by the cloud server are placed into the cluster head node is done during the initialization of the cluster head node. The base station then transmits its corresponding $\hat{T}_n$ to cluster head node n within its signal coverage and cluster head node n, after receiving the corresponding key exchange coupling synapse matrix $\hat{T}_n$ transmitted by the base station, calculates:

$$\hat{T}_{total} = H_n * \hat{T}_n * H'_n = H_a * H_b * H_c * \ldots * H_n * H * T * H' * H'_n * \ldots * H'_c * H'_b * H'_a \tag{8}$$

Once all cluster head nodes have derived $\hat{T}_{total}$, the shared key between the base station layer and the cluster head layer within its signal coverage is established. For groups outside the base station signal, the key exchange

between the cluster head node and the base station requires the routing node to pass the key exchange synapse matrix corresponding to cluster head node n $\hat{T}_n$, and the T disclosed by the base station to that cluster head node. The flow is shown in Fig. 5.

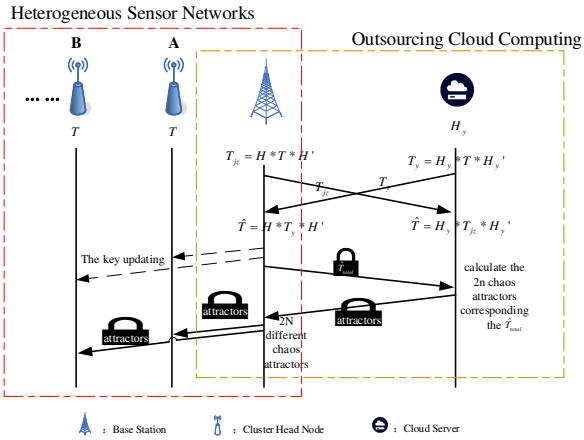

Fig. 5 Establishment of shared keys at the base station level and cluster head level

4.3.2 Establishment of shared keys between the cluster head layer and the perception layer

At deployment time, private keys of different key space sizes H_n are initialized for different groups according to the security level of the group, and are placed at the initialization of the unique cluster head node corresponding to the sensor node. The cluster head node intercepts the first m columns of the first row of its original n-order private key H_n to form a new m-order rotation matrix H_n according to the security level of the group, and at the same time selects the corresponding order of the coupling synapse matrix T in the starting pool of coupling synapse matrices and exposes it, as shown in Fig. 6. Then, the shared key $\hat{T}_{(a)total}$ and the key exchange synapse matrix for each sensor node are calculated as follows:

$$\hat{T}_{(a)total} = H_{(a)1} * H_{(a)2} * \ldots * H_{(a)n} * H_a * T * H'_a * H'_{(a)n} * \ldots * H'_{(a)2} * H'_{(a)1}$$
$$\hat{T}_{(a)1} = H_{(a)2} * \ldots * H_{(a)n} * H_a * T * H'_a * H'_{(a)n} * \ldots * H'_{(a)2}$$
$$\hat{T}_{(a)2} = H_{(a)1} * \ldots * H_{(a)n} * H_a * T * H'_a * H'_{(a)n} * \ldots * H'_{(a)1}$$
$$\ldots$$
$$\hat{T}_{(a)n} = H_{(a)1} * H_{(a)2} * \ldots * H_{(a)n-1} * H_a * T * H'_a * H'_{(a)n-1} * \ldots * H'_{(a)2} * H'_{(a)1} \tag{9}$$

where $\hat{T}_{(a)1}, \hat{T}_{(a)1}, \ldots, \hat{T}_{(a)n}$ is the key exchange synapse matrix of the sensor node $1, 2, 3, \ldots, n$ respectively.

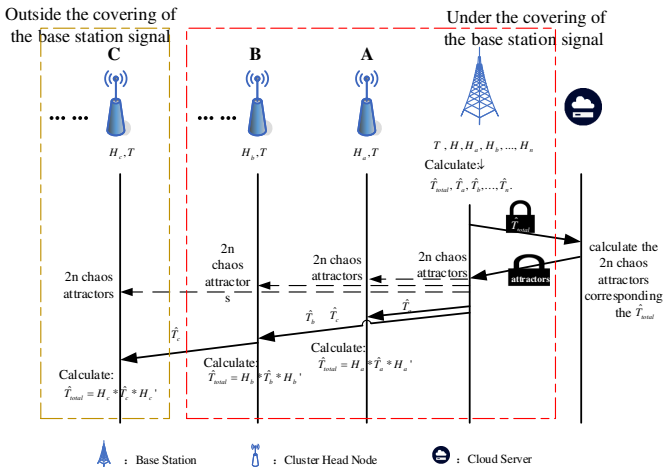

Fig. 6 Correspondence between group safety levels and the matrix of linked synapses

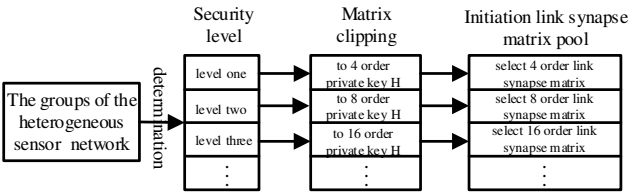

Fig. 7 Establishment of shared keys between the cluster head layer and the perception layer

The cluster head node transmits its corresponding $\hat{T}_{(a)n}$ to the sensor node n in its group and encrypts $\hat{T}_{(a)total}$ with a sequence of pseudo-random numbers generated by the shared key substitution algorithm at the cluster head layer and transmits it to the base station layer. The base station transmits $\hat{T}_{(a)total}$ to the cloud server to calculate the chaotic attractor corresponding to $\hat{T}_{(a)total}$, and then transmits it to the cluster head node via the base station after the cloud server has finished calculating it and sets the attractor in the sensor node during its initialization. After receiving the corresponding $\hat{T}_{(a)n}$, the sensor node calculate:

$$\begin{aligned} \hat{T}_{(a)total} &= H_{(a)n} * \hat{T}_{(a)n} * H'_{(a)n} \\ &= H_{(a)1} * H_{(a)2} * \ldots * H_{(a)n} * H_a * T * H'_a * H'_{(a)n} * \ldots * H'_{(a)2} * H'_{(a)1} \end{aligned} \tag{10}$$

All sensor nodes calculate $\hat{T}_{(a)total}$, the shared key establishment between cluster head layer and sensing layer is completed and the process is shown in Fig. 7.

4.4 Continuous validation

This paper relies on CSI to update keys periodically and provides a dynamic feature to ensure continuous verification.

4.4.1 Perceptual node continuity verification

The process is as follows:

(i)After the shared key is established, the cluster head node calculates the CSI of the data packet received from the sensor node, i.e. C_E . In order to avoid replay attack and determine the freshness of the data packet, we also add counters C_g and C_e . The cluster head node sets C_g to 1, and encrypts and transmits C_E and random number r to the sensor node with the public key PK_E of the sensor node.

(ii)After receiving the data packet, the sensor node decrypts it with its private key to obtain the CSI values C_E and r , sets C_e to 1, randomly selects an index value (i.e."a"), obtains ma through the XOR of a and the hash value of the XOR value of the shared keys $\hat{T}_{(a)total}$ and$r$, sets the hash value of C_E as the verification key V_{key} for this verification, and transmits the ma value to the cluster head node through this key, At the same time, the $hash$ value of this data transfer is also up-linked.

(iii)After receiving the data packet, the cluster head node first queries the hash value on the chain to prevent message tampering. After it is correct, the verification key decrypts the message to obtain the ma value, deduces the a' value by calculating the XOR value of the hash value of the XOR between ma and the shared key and random number r , and calculates the verification time difference t through the timestamp $t_c - t_s$. If it is greater than the set upper limit value T , the verification is invalid, and the ACK value of the

control verification interrupt is set to 0, jump back to the authentication stage again, otherwise, set it to 1 to continue the authentication. The cluster head node then continues to randomly set the second index (i.e."b"), adds one to the local counter C_g , calculates the mb through the XOR of the hash value of the XOR value of b and $\hat{T}_{(a)total}$ and r , and transmits $[ACK, t, mb, a', C_g]$ encryption to the sensor node through V_{key} .

(iv)The sensor node receives the data, decrypts the data with the authentication key, verifies the ACK value, and returns to mutual authentication if the value is 0. If $ACK = 1$ and $a' = a, C_e + 1 = C_g$, continue the verification. The b' is obtained by the XOR of the hash value of the XOR of mb , $\hat{T}_{(a)total}$ and r , and the local counter C_e is incremented by one. The C_e is encrypted and transmitted to the cluster head node with the verification key, and the message is chained.

(v)The cluster head node receives the hash value of the data uplink query, decrypts the message with the verification key and verifies the $C_g = C_e$. If the time difference t is less than T , the verification is successful, and then repeat the same process to complete the continuous verification of the sensor node. The process is shown in Fig. 8.

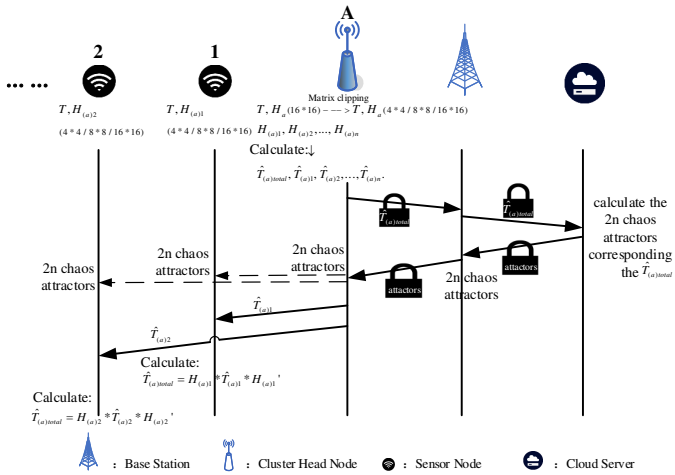

Fig. 8 Continuous verification process for sensor nodes

4.4.2 Continuous verification of cluster head nodes

The process is the same as the continuous verification process of the sensor nodes. The base station will perform continuous verification of the cluster head node, and if the authentication fails, it will jump back to the authentication

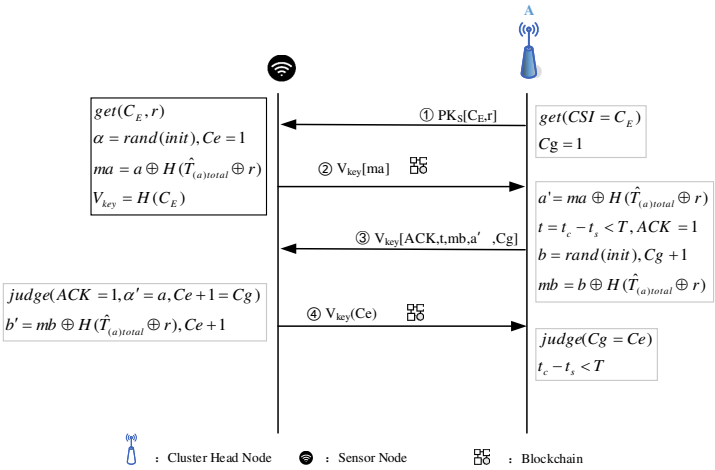

Fig. 9 Continuous verification process for cluster head nodes

process, and the sensor nodes in the group will also re-authenticate and re-establish the shared key. The process is illustrated in Fig. 9.

4.5 Exit of nodes

4.5.1 Exit of cluster head nodes

Due to the differences in communication and routing protocols used by the network, after a cluster head node exits, the cluster-aware nodes of the group may be merged into other groups or a new cluster head node may be elected, or all the nodes of the group may fail, depending on the specific network. The key management protocol focuses on how to secure the network after a node exits. Specifically, after a cluster head node exits, the base station removes the identity information of the node from the blockchain's legal list and the private key of the group's unique cluster head node stored in the base station at H_n . The cluster head node of the group requesting to exit can no longer access the network information through the identity authentication process.

4.5.2 Perception node exit

In HSN, if a sensor node in a group exits the sensing network, the base station will delete the identity information of the node in the blockchain legal list and send an exit command to the cluster head node of the corresponding group. After receiving the command, the cluster head node deletes the private key of the requesting sensor node stored in the node, and the requesting sensor node can no longer access the in-network information through the continuous verification process.

5 Experimental analysis and theoretical proof

5.1 Instance analysis of determination of security level

Suppose take a determination of security level for a group of the heterogeneous sensor network.

1.Determine $I_i(i = 1, 2, 3, 4)$, I_1 = computing power of the node, I_2 = storage capacity, I_3 = group security requirements, I_4 = environment the node locates.

2.Determine judge grade domain of discourse: $L(L_1, L_2, L_3, L_4)$, where L_1 is security level 1, i.e. the lowest security level; L_2 is security level 2, by parity of reasoning, L_4 is security level 4, the highest security level.

3.According to Table 1, each expert makes marking for each evaluation element, then $\bar{x_i}$ is derived by three points estimation method [32] and the normalization processing is taken. The weight vector expression of determining the evaluation factor is as following: $W = (w_1 w_2 w_3 w_4) = (0.14300.17090.36440.3217)$.

4.According to Table 2, the expert makes marking and establishes the relation matrix of degree of membership.

$$R = \begin{pmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{pmatrix}_{ij} = \begin{pmatrix} 0.9668 & 0.9303 & 0.8512 & 0.7763 \\ 0.9640 & 0.9186 & 0.7974 & 0.7087 \\ 0.5350 & 0.6115 & 0.8719 & 0.7983 \\ 0.5450 & 0.6583 & 0.7562 & 0.7183 \end{pmatrix} \tag{11}$$

5.Synthesis synthetic evaluation model. For judge the security level of groups of the heterogeneous sensor networks, our paper employ the evaluation model [32] and synthesis the weight vector W of evaluation factor with the relation matrix R of the thing evaluated and get the synthetic evaluation result vector A of the thing evaluated.

$$\begin{aligned} A &= W \cdot R \\ &= \begin{pmatrix} 0.1430 & 0.1709 & 0.3644 & 0.3217 \end{pmatrix}. \\ &= \begin{pmatrix} 0.9668 & 0.9303 & 0.8512 & 0.7763 \\ 0.9640 & 0.9186 & 0.7974 & 0.7087 \\ 0.5350 & 0.6115 & 0.8719 & 0.7983 \\ 0.5450 & 0.6583 & 0.7562 & 0.7183 \end{pmatrix} \end{aligned} \tag{12}$$

From the maximum membership principle it can be seen that the security level of that groups of the heterogeneous sensor networks is security level 3.

5.2 Security analysis

Firstly, the algorithm security is analysed.The security of the neural network chaos encryption algorithm is based on the difficulty of decomposing the singular matrix and the chaotic classification property of the Hopfield

Table 1 The weight information of index factor

	Expert 1	Expert 2	Expert 3	Expert 4	$\bar{x}_i$	w_i
Index factor 1	0.10	0.15	0.17	0.17	0.1422	0.1430
Index factor 2	0.20	0.15	0.15	0.18	0.1700	0.1709
Index factor 3	0.40	0.35	0.38	0.32	0.3625	0.3644
Index factor 4	0.30	0.35	0.30	0.33	0.3200	0.3217

overloaded chaotic neural network. For the encryption algorithm employed in KMPHSN-ZTSCNN, the key can be attacked based on the classification property of the Hopfield overloaded chaotic neural network or by decomposing the singular matrix to perform the key search. The paper [34] demonstrates that the algorithm is resistant to both attacks. On the other hand, the security of the protocol is analysed in terms of its performance metrics, which include forward security, backward security and resistance to complicity. These three metrics are clearly defined in the paper [35]. According to the definitions it can be proved that the KMPHSN-ZTSCNN protocol proposed in this paper is secure.

Lemma 1 KMPHSN-ZTSCNN is the forward secure.

PROVING

In KMPHSN-ZTSCNN, when a cluster head node leaves the current layer or group, the base station will delete the identity information of the node in the blockchain legal list and the private key of the corresponding cluster head node stored in the base station H_n ; when a sensor node leaves the current group, the base station will delete the identity information of the node in the blockchain legal list and notify the cluster head node to delete the private key of the sensor node requesting to exit. As a result, after the cluster head node and the sensor node exit the current layer or group, their identity becomes non-member nodes and they cannot pass the continuous verification process, so they cannot obtain the information in the network and the forward security of the network is ensured. The protocol is forward-secure and proves completion.

Lemma 2 KMPHSN-ZTSCNN is the backward secure.

PROVING

In KMPHSN-ZTSCNN, when a cluster head node joins the sensor network, it first needs to shake hands with the base station. After confirming the object, the base station assigns an certificate of identity to it, changes its coupling synapse matrix T , and recalculates the shared key $\hat{T}_{(a)total}$ and the key exchange coupling synapse matrix $\hat{T}_a, \hat{T}_b, \hat{T}_c, \ldots$ for each cluster head node, and then performs a key exchange with each cluster head node. This ensures that the key exchange synapse matrix $\hat{T}_n$ for all key exchanges with cluster head nodes is not the shared key of the previous layer or group, even if a node has been physically captured before, and that its key cannot be decrypted. Also, a newly added cluster head node cannot compute the shared key of any previous layer or group in the network at any point in time. The protocol is backward secure and proves completion.

Table 2 Statistical information of evaluation index

Class	Security level 1				Security level 2			
	Expert 1	Expert 2	Expert 3	Expert 4	Expert 1	Expert 2	Expert 3	Expert 4
Index factor 1	[95,98]	[96,97]	[95,99]	[95,98]	[90,95]	[92,95]	[90,93]	[95,97]
Index factor 2	[95,98]	[94,97]	[95,98]	[98,99]	[92,95]	[90,92]	[88,92]	[93,95]
Index factor 3	[50,60]	[55,60]	[45,50]	[50,55]	[65,68]	[60,62]	[58,60]	[55,65]
Index factor 4	[50,60]	[55,60]	[50,55]	[50,55]	[65,68]	[60,65]	[65,70]	[68,70]
Class	Security level 3				Security level 4			
	Expert 1	Expert 2	Expert 3	Expert 4	Expert 1	Expert 2	Expert 3	Expert 4
Index factor 1	[80,85]	[85,87]	[83,87]	[90,92]	[78,82]	[75,80]	[72,77]	[80,82]
Index factor 2	[75,80]	[77,80]	[78,82]	[80,85]	[65,72]	[70,72]	[70,75]	[70,75]
Index factor 3	[85,90]	[82,88]	[87,90]	[90,92]	[75,80]	[80,82]	[77,80]	[80,85]
Index factor 4	[75,80]	[72,77]	[68,70]	[75,80]	[70,73]	[68,70]	[70,75]	[70,75]

Lemma 3 KMPHSN-ZTSCNN can resist collusion attack.
PROVING

In KMPHSN-ZTSCNN, a node can only obtain its own private keyand the key H_n exchange link synapse $\hat{T}_n$ to exchange keys with the base station. Even if multiple legitimate nodes conspire, they cannot compute the private keys of other nodes associated with the system, nor can they compromise the system. Thus, the protocol is resistant to conspiracy attacks, and proves completion.
Lemma 4 KMPHSN-ZTSCNN can achieve zero trust security.
PROVING

In KMPHSN-ZTSCNN, a node has to enter the continuous verification process after joining the network and completing the establishment of a shared key, and once the node is maliciously attacked or physically captured during the period, the continuous verification process will be interrupted and jump back to the authentication process again to ensure that the wireless sensor network can achieve secure data transmission in a zero-trust environment. The protocol is capable of zero-trust security, and proves completion.

5.3 KMPHSN-ZTSCNN effectiveness analysis

Lemma 5 KMPHSN-ZTSCNN guarantees packet freshness
PROVING

In KMPHSN-ZTSCNN, counters C_g and C_e are added to successive verifications, and the value of the counters is judged in each verification to avoid replay attacks. A timestamp $t_c - t_s$ is also added to calculate the time difference t of the verification and compare it with the set threshold T to ensure that the packets are transmitted within the specified time and to ensure the freshness of the packets. Once the value of the counter or the time difference in the authentication overage process is large, the authentication is terminated and the authentication process is reverted back to the end of the certificate,and proves completion.
Lemma 6 KMPHSN-ZTSCNN guarantees data traceability
PROVING

When a node joins the network, it has the identity in the legitimate list of the blockchain, and the node needs a private key signature to upload data, ensuring that the source of each piece of data can be traced, and the location of the malicious node can be quickly located. The hash value of the data packets in the continuous verification will also be up-chained to ensure that the data is traceable and cannot be modified by the fact that the hash value will change if there is any change to the data and the hash function calculation process is one-way irreversible,and proves completion.

5.4 Comparison of KMPHSN-ZTSCNN with other key management schemes

As shown in Table 3, some of the KMPHSN-ZTSCNN nodes have pools of coupled synaptic matrices. Since the KMPHSN-ZTSCNN protocol supports

Table 3 Comparison of KMPHSN-ZTSCNN with other key management schemes

Scheme and protocols	Structure of key pool	Scalability	Key connectivity	traceability
E-G scheme [36]	Non-Structure	Good	$1-\frac{((p-k)!)^2}{(p-2k)!p!}$	No
Multiple-Space key pre-distribution scheme [37]	Structured	Weak	$1-\frac{((\omega-t)!)^2}{(\omega-2t)!\omega!}$	No
CPKS [38]	Structured	Good	$\frac{c}{m}\iint_{(x-i_x)^2+(y-i_y)^2\leq d^2}\frac{p\left(v_{jx,jy},u_{ix,iy}\right)}{\pi d'^2}dxdy$	No
LBKP [38]	Structured	Moderate	$\frac{\sum_{c_{jc,jr}\in s_{c,ir}}p(C_{jc,jr},C_{ic,ir})}{\sum_{\forall j_c,j_r}p(C_{jc,jr},C_{ic,ir})}$	No
Grid-Group deployment scheme [39]	Structured	Moderate	$1-\frac{((\omega-t)!)^2}{(\omega-2t)!\omega!}$	No
M-IBE Based Key Management Protocol for Heterogeneous Sensor Networks [32]	Structured	Good	1	No
KMPHSN-ZTSCNN	Structured	Good	0.17	Yes

dynamic node exit and join, KMPHSN-ZTSCNN has better scalability. In addition, because of the addition of blockchain storage of node identity information and node behaviour in this protocol, there is better traceability of node behaviour. In addition, any two legitimate nodes are able to establish a shared key through key exchange, and therefore the connectivity law of the proposed protocol is constant to 1.

6 Conclusion

This paper proposes a key management protocol for heterogeneous sensor networks based on zero trust security and chaotic neural networks. The protocol judges the security level of groups in HSNs, allocates key spaces of different sizes to groups of different security levels, and performs hierarchical management and group management of shared keys. The protocol solves the problem that sensor nodes in HSN can only use fixed identical key spaces to calculate and encrypt keys under different storage spaces and different computing capacities. In addition, the protocol proposes a scheme to outsource chaotic attractor computation to a cloud computing platform and an authentication scheme combining blockchain and zero trust, which reduces the computational complexity of network nodes while improving security; avoids the repeated computation of chaotic attractors on cryptographic communication network nodes and simplifies the authentication algorithm.

The performance analysis of the proposed protocol shows that it is highly secure and has good scalability and connectivity. In terms of effectiveness, the protocol's computational overhead and storage requirements show a significant hierarchical grouping that can accommodate the heterogeneous elements of the HSN and allow the high-energy nodes to function efficiently. This paper provides a new solution for the application of zero-trust based security and chaotic neural networks in HSNs.

Acknowledgments This work was supported by the National Natural Science Foundation of China (NO.61370007). The authors wish also to thank the anonymous reviewers for their comments.

References

[1] Du, X.,Xiao,Y.,Guizani,M.,Chen,H.H.:An effective key management scheme for heterogeneous sensor networks. Ad hoc networks,5(1),24-34(2007)

[2] Girod,L.,Stathopoulos,T.,Ramanathan,N.,Elson,J.,Estrin,D.,Osterweil, E.,Schoellhammer,T.:A system for simulation, emulation, and deployment of heterogeneous sensor networks.In Proceedings of the 2nd international conference on Embedded networked sensor systems ,pp.201-213.(2004)

[3] Walters,J.P.,Liang,Z.,Shi,W.,Chaudhary,V.:Wireless sensor network security: A survey. In Security in distributed, grid, mobile,and pervasive

computing ,pp.367-409.Auerbach Publications.(2007)

[4] Parenreng,J.M.:A model of security adaptation for limited resources in wireless sensor network. Journal of Computer and Communications, 5(03), 10.(2017)

[5] Juric,R.,Lyth,A.,Larson, D.:Group Key Management in Wireless Sensor Networks: Introducing Context for Managing the Re-keying Process.In Proceedings of the 55th Hawaii International Conference on System Sciences.(2022)

[6] Menezes,A.J.,Van Oorschot,P.C.,Vanstone,S.A.:Handbook of applied cryptography.CRC press.(2018)

[7] Al-Riyami,S.S.,Paterson,K.G.:Certificateless public key cryptography. In International conference on the theory and application of cryptology and information security ,pp. 452-473,Berlin,Heidelberg.(2003).Springer

[8] Eltoweissy,M.,Moharrum,M.,Mukkamala R.:Dynamic key management in sensor networks. IEEE Communications magazine, 44(4), 122-130.(2006)

[9] Zhang, W.,Zhu,S.,Cao,G.:Predistribution and local collaboration-based group rekeying for wireless sensor networks.Ad hoc networks,7(6), 1229-1242.(2009)

[10] Divya,R.,Thirumurugan,T.:A novel dynamic key management scheme based on hamming distance for wireless sensor networks.In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) ,pp.181-185.(2011).IEEE.

[11] Rahman,S.M.M.,El-Khatib,K.:Private key agreement and secure communication for heterogeneous sensor networks. Journal of Parallel and Distributed Computing,70(8), 858-870.(2010)

[12] Chatterjee,K.,De,A.,Gupta,D.:An improved ID-based key management scheme in wireless sensor network. In International Conference in Swarm Intelligence,pp. 351-359,Berlin, Heidelberg.(2012).Springer,

[13] Seo, S.H.,Won,J.,Sultana,S.,Bertino,E.:Effective key management in dynamic wireless sensor networks. IEEE Transactions on Information Forensics and Security, 10(2), 371-383.(2014)

[14] Tian,Y.,Wang,Z.,Xiong, J.,Ma,J.:A blockchain-based secure key management scheme with trustworthiness in DWSNs. IEEE Transactions on Industrial Informatics, 16(9), 6193-6202.(2020)

[15] Truong,H.T.T.,Almeida,M.,Karame,G.,Soriente,C.:Towards secure and decentralized sharing of IoT data. In 2019 IEEE International Conference on Blockchain (Blockchain),pp. 176-183.(2019).IEEE

[16] Kindervag,J.:Build security into your network's dna: The zero trust network architecture. Forrester Research Inc, 27.(2010)

[17] Yan,X.,Wang, H.:Survey on zero-trust network security. In International Conference on Artificial Intelligence and Security ,pp.50-60, Singapore.(2020).Springer

[18] Mehraj,S.,Banday,M.T.:Establishing a zero trust strategy in cloud computing environment. In 2020 International Conference on Computer Communication and Informatics (ICCCI) ,pp.1-6.(2020).IEEE

[19] Omar,R.R.,Abdelaziz,T.M.:A comparative study of network access control and software-defined perimeter. In Proceedings of the 6th International Conference on Engineering & MIS 2020,pp. 1-5.(2020)

[20] Kumar,P.,Moubayed,A.,Refaey,A.,Shami,A.,Koilpillai, J.:Performance analysis of sdp for secure internal enterprises. In 2019 IEEE Wireless Communications and Networking Conference (WCNC),pp. 1-6.(2019).IEEE

[21] Albuali,A.,Mengistu,T.,Che, D.:ZTIMM: A zero-trust-based identity management model for volunteer cloud computing. In International Conference on Cloud Computing ,pp. 287-294.Cham,(2020).Springer

[22] Balfour,R.E.:Building the "Internet of Everything"(IoE) for first responders. In 2015 Long Island Systems, Applications and Technology ,pp.1-6.(2015).IEEE

[23] Sateesh,H.,Zavarsky, P.:State-of-the-Art VANET trust models: challenges and recommendations. In 2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON),pp. 0757-0764.(2020).IEEE

[24] Chen,Y.,Hu,H.C.,Cheng,G. Z.:Design and implementation of a novel enterprise network defense system bymaneuveringmulti-dimensional network properties. Frontiers of Information Technology & Electronic Engineering, 20(2), 238-252.(2019)

[25] Dimitrakos,T.,Dilshener,T.,Kravtsov,A.,La Marra,A.,Martinelli,F.,Rizos, A.,Saracino,A.:Trust aware continuous authorization for zero trust in consumer internet of things. In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom),pp. 1801-1812.(2020).IEEE

[26] Puthal,D.,Mohanty,S.P.,Nanda,P.,Choppali,U.:Building security perimeters to protect network systems against cyber threats [future directions]. IEEE Consumer Electronics Magazine, 6(4), 24-27.(2017)

[27] Zaheer,Z.,Chang,H.,Mukherjee,S.,Van der Merwe,J.:eztrust: Network-independent zero-trust perimeterization for microservices. In Proceedings of the 2019 ACM Symposium on SDN Research ,pp. 49-61.(2019)

[28] Chen,Z.,Yan,L.,Lü, Z.,Zhang,Y.,Guo,Y.,Liu,W.,Xuan, J.:Research on zero-trust security protection technology of power IoT based on blockchain. In Journal of Physics: Conference Series ,Vol. 1769, No. 1, p. 012039. IOP Publishing.(2021)

[29] Chifor,B.C.,Arseni,S.C.,Matei,I.,Bica,I.:Security-oriented framework for internet of things smart-home applications. In 2019 22nd International Conference on Control Systems and Computer Science (CSCS),pp. 146-153.(2019).IEEE

[30] Samaniego,M.,Deters, R.:Zero-trust hierarchical management in IoT. In 2018 IEEE international congress on Internet of Things (ICIOT) ,pp. 88-95.(2018).IEEE

[31] Sultana,M.,Hossain,A.,Laila,F.,Taher,K.A.,Islam,M.N.:Towards developing a secure medical image sharing system based on zero trust principles and blockchain technology. BMC Medical Informatics and Decision Making, 20(1), 1-10.(2020)

[32] Liu,H.C.,You,J.X.,You,X.Y.,Shan,M.M.:A novel approach for failure mode and effects analysis using combination weighting and fuzzy VIKOR method. Applied soft computing, 28, 579-588.(2015)

[33] Luo,Y.,Guo,H.,Zhang,L.T.,Cai,A.,Gui, N.:Grey reliability design model and its application to mechanical engineering. Journal of National University of Defense Technology(China), 24(1), 94-99.(2002)

[34] Guo-Gang,L.,Dong-Hui,G.:One-way property proof in public key cryptography based on OHNN. Procedia Engineering, 15, 1812-1816.(2011)

[35] Sastry,N.,Wagner,D.:Security considerations for IEEE 802.15. 4 networks. In Proceedings of the 3rd ACM workshop on Wireless security,pp. 32-42.(2004)

[36] Eschenauer, Laurent, Gligor, et al. A key-management scheme for distributed sensor networks[J]. 2002

[37] Eschenauer,L.,Gligor,V.D.:A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM Conference on Computer

and Communications Security ,pp. 41-47.(2002)

[38] Liu,D.,Ning,P.:Location-based pairwise key establishments for static sensor networks. In Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks ,pp.72-82.(2003)

[39] Huang,D.,Mehta,M.,Medhi,D.,Harn,L.:Location-aware key management scheme for wireless sensor networks. In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks ,pp. 29-42.(2004)