Research Article

A Hierarchical Matrix Decomposition-Based Signcryption without Key-Recovery in Large-Scale WSN

Chi Yuan , Wenping Chen, and Deying Li

School of Information, Renmin University of China, China

Correspondence should be addressed to Deying Li; deyingli@ruc.edu.cn

Received 16 July 2018; Revised 19 October 2018; Accepted 13 November 2018; Published 2 December 2018

Guest Editor: Georgios Kambourakis

Copyright © 2018 Chi Yuan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The sensors in wireless sensor network (WSN) are vulnerable to malicious attacks due to the transmission nature of wireless media. Secure and authenticated message delivery with low energy consumption is one of the major aims in WSN. The identity-based key authentication scheme is more suitable for the WSN. In this paper, the Hierarchical Matrix Decomposition-based Signcryption (HMDS) algorithm was proposed, which is a kind of identity-based authentication scheme. In HMDS scheme, three-layer architecture, base station (BS), cluster head, and intracluster, is employed to adapt to the common structure of WSN. As the key generation center (KGC), the BS adopts matrix decomposition to generate the identification information and public key for cluster head, which not only reduces the cost of calculation and storage but also avoids the collusion attack. Experiments show that the HMDS algorithm has more advantages over other algorithms and is very suitable for the large-scale WSN.

1. Introduction

Wireless sensor network (WSN) combined sensor and network communication technology has real-time sensing and information acquisition functions. It is applied to various fields such as national defense, environment monitoring, transportation management, medical treatment, and public health [1, 2]. The wireless sensor network has the characteristics of large-scale, multihop communication and complex deployment environment, and so forth [3, 4]. At the same time, since the sensor network nodes mostly have small size and are deployed in unattended harsh environment, the sensor node energy as well as the computing capability is greatly limited. Therefore, low energy cost and computation complexity are important requirements for WSN application [5]. In addition, WSN is different from the wired network, the communication channel of wireless network is public, and it is not difficult for an adversary to manipulate the sensors in an unprotected WSN. As a result, the sensors in a WSN are vulnerable to malicious attacks. Thus, the energy efficiency and network security must be considered when designing WSN application [6–9].

Cryptography plays a very important role in security [10–12]. The common public key encryption methods can be

classified into three categories: Public Key Infrastructure (PKI) based encryption, identity-based encryption (IBE) and the certificate-free encryption [13]. Among them, the identity-based encryption (IBE) method has obvious advantage in both security and calculation complexity. The literature [14] pointed out that the IBE method is the most suitable public key system for wireless sensor networks (WSN). However, there is a flaw in the key escrow scheme of IBE algorithms. Because the Private Key Generator (PKG) holds private keys of all users, it can easily impersonate any node user, decrypt its ciphertext, and forge user signatures.

In this paper, the Hierarchical Matrix Decomposition-based Signcryption (HMDS) algorithm was proposed to resolve the above problems. In HMDS algorithm, the clustering management scheme is employed to adapt to the data aggregation architecture in WSN. The matrix decomposition-based method is adopted to generate the keys for cluster head nodes, which makes use of identification information and avoids absolute control of KGC on the private keys. Such mechanism really solves the key escrow problem in IBE algorithm. To evaluate the performance of the HMDS scheme, we compare it with the Hierarchical Identity-Based Signcryption Scheme (HIS) algorithm [15]. The experiment results show that the HMDS algorithm is more suitable

for the WSN. Especially in the large-scale WSN, the HMDS algorithm is very stable and has advantage over HIS scheme.

2. Related Works

2.1. Public Key Authentication. In recent years, many researchers have done a lot of work on the key authentication for WSN. There are three kinds of authentication methods, PKI-based encryption, identity-based encryption, and certificate-free encryption.

2.1.1. PKI-Based Encryption. In a typical PKI-based scheme, the public key certificate is signed by a certificate authority (CA) to achieve the binding between the user's public key and its identity. The certificate-based key management scheme is the most common authentication scheme. However, the PKI-based key management scheme tends to consume too much storage, calculation, and communication resource, which is unaffordable for the energy-constrained sensor nodes in WSN [16, 17].

2.1.2. Identity-Based Encryption. In 1984, the identity-based encryption algorithm was first proposed by Shamir [18], which enabled any pair of users to communicate securely and verify each other's signatures without exchanging private or public keys. In 2003, Chen studied the key management protocol in identity-based encryption scheme [19]. In order to simplify the management of the key, the user's public key was used for directly calculating the user's identity information, and the private key was credibly generated by PKG (Private Key Generator). Without the public key certificate, such scheme reduced the storage and computing cost for issuance, cancellation, and certification.

To reduce the energy consumption of the identity-based authentication node, some identity-based key management protocols for WSN were proposed [20, 21]. The literature [11] pointed out that the identity-based encryption was the most suitable public key scheme for wireless sensor networks (WSN). Neal Koblit [22] and Victor Miller [23] proposed elliptic curve cryptosystem. Then, the similar algorithm was proposed by Zhang et al. in [24], which presented that the security of ECC was based on the discrete logarithm problem. Compared with other public key systems, this scheme is a promising method with high security, low cost, and high efficiency. However, in most elliptic curves based key management schemes, grouping management mechanism was not introduced. With the increasing of nodes, more communication with BS is needed, which was not suitable for the nodes with limited energy in WSN.

In the key-insulated cryptography proposed by Qin Zhiguang et al., the key was divided into two parts: one part was managed by the users and the other part was saved by a physical security helper [12]. When the key was needed, the two parts of the key were spliced into a complete key. But it failed to take advantage of the identity information from each effective node. Chen Yuan et al. [13] proposed an identity-based encryption scheme without bilinear pairings.

Both algorithms used the idea of noncertificate; the private key was generated by the node and the Private Key Generator (PKG). Therefore, the two algorithms could not really solve the key escrow problem.

Guo Jianghong et al. [16] proposed a new key agreement scheme for WSN. The node established the pairing key through the Diffie-Hellman protocol, and the required key parameters are obtained through broadcast. The scheme still has the advantages of the identity-based encrypted key agreement scheme, but the time and energy consumption for bilinear pairing operation is very high. Guo et al. [25] combined the identity-based RSA mechanism with the lightweight Certificate Authority (CA) to construct an identity-based mixed model, ECC-CA, for sensor network cryptography scheme. But the scheme has relatively huge cost of calculation and communication.

2.1.3. Certificate-Free Encryption. To resolve the certificate problem in the PKI-based scheme and the key escrow problem in the identity-based scheme, the certificate-free encryption method was proposed. In certificate-free encryption algorithm, a credible third-party Key Generation Center (KGC) was still employed to jointly generate the user's private key. KGC did not directly hold the users' private keys but only generated a partial private key. The users themselves generated the final private key instead. In such scheme, the public key information could not be directly obtained from the user identity information [26, 27]. At present, these provable security certificate-free construction methods are based on the Waters hash function, which leads to long system parameters and requires more pairing calculations. Thus, these methods are not suitable for WSN with limited storage capacity and calculation capability.

2.2. Cluster-Based Authentication for WSNs. Zhang et al. [28] used two cluster heads to realize the system authentication work. They assumed that the cluster head (CH) had limited processing capacity and could not meet the requirements of data calculation. In a cluster, one CH was responsible for intercluster communication and the other CH was responsible for the collection and processing of data in its cluster. Dai et al. [29] also proposed a dual cluster head authentication scheme, whose main principle was similar to the system proposed by Zhang. The main CH and the sub-CH were selected in the larger clusters; the main CH was assisted by the sub-CH. Wang et al. [30] proposed a nonuniform clustering management scheme achieved by establishing clusters with different sizes, which was different from dual cluster head structure. However, it did not consider the fact that the influence of fixed factors on network varies with the number of rounds. Based on the dynamic search strategy, in literature [31], dynamic CH was introduced to collect the interference data of illegal nodes, but it could not realize adaptive switching data collection path. Yu et al. [32] proposed a DEER algorithm to find the optimal path between a CH and the coordinator, but it may lead to unbalanced load among the CHs.

FIGURE 1: Architecture of HMDS.

Rohbanina et al. [33] proposed a hierarchical WSN for key management, which first constructed the shortest path and uses the elliptic curve-based cryptography scheme for session key distribution. DENG et al. [34] made use of the network differentiation to improve system communication efficiency, which adopted the grouping encryption algorithm. By using the summary information comparison method and the singular point exclusion strategy, the system recoverability was enhanced. Klaoudatou et al. [35] thoroughly evaluated the cluster-based Group Key Agreement (GKA) protocols for WSNs. The authors examined many related literatures to study their performance and energy consumption. They thought that clustering is ideal for large-scale environments and time-critical applications. The use of cluster-based approaches optimizes network bandwidth and service discovery while addressing the needs for scalability at the same time.

3. Hierarchical Matrix Decomposition-Based Signcryption (HMDS) Scheme

3.1. Overview of HMDS Scheme. In the WSN, cluster-based hierarchical architecture could decrease the communication overhead of data collection and aggregation. Moreover, it is good to extend the lifetime if nodes are assigned to different roles according to their resources. Therefore, we propose a Hierarchical Matrix Decomposition-based Signcryption (HMDS) scheme, which adopts the cluster-based hierarchical architecture illustrated as in Figure 1. There are three layers, base station (BS), cluster head layer, and intracluster layer. Usually, the WSN consists of a few clusters and a BS. When a node communicates with other nodes in different cluster, the message must be forwarded by the corresponding CHs.

In the HMDS scheme, the BS is used as KGC. There are three modules for performing the KGC functions, database

module, encryption machine module, and key management system module. The BS assigns an identification number and generates the public key for each CH in cluster head layer. Matrix D is stored in the database on BS. Given a symmetric matrix D , it is Doolittle-decomposed into two triangular matrices L and R ; thus, the ID of each node can be assigned quickly according to the polynomial function. Furthermore, the authentication and key updating of the node can be realized.

The nodes with more storage and computing resource are assigned as CHs. Each CH is responsible for managing a number of ordinary sensor nodes and performing more complex operations. In the cluster head layer, each CH computes the public key for each sensor node in its cluster.

Before presenting the detail of the HMDS scheme, it is necessary to introduce some deployed notations as described in Table 1.

3.2. Signcryption for Intercluster Communication

3.2.1. Initialization. The BS, as the KGC, is trustable to generate key for CHs. Before generating keys, the steps of initialization are described as follows.

(A) BS selects a $(n \times n)$ symmetric matrix D and saves it. Each order principal minor determinant of D is not equal to 0; the matrix D is represented as follows:

$$D = \begin{bmatrix} d_{11} & & & \\ & \ddots & & \\ & & d_{ij} & \\ & & d_{ji} & \ddots \\ & & & & d_{nn} \end{bmatrix} \quad (1)$$

where $d_{ij} = d_{ji}$.

TABLE 1: Deployed notations.

Notations Guide	
D	(n×n) symmetric matrix
L	The lower triangular matrix of Doolittle-decomposed for matrix D
R	The upper triangular matrix of Doolittle-decomposed for matrix D
S_i/P_i	Secret key/ Public key of applicant ID_i
Id_i	Identification number of cluster head node i
G	The base point on the elliptic curve
U_i	The verification parameter of node i
H	Hash function

(B) As the KGC, the BS performs Doolittle decomposition for the matrix D as follows:

$$D = L \cdot R$$

$$\begin{aligned}
 &= \begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \dots & \dots & \dots & \\ l_{n,1} & l_{n,2} & \dots & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ & r_{2,2} & \dots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{bmatrix} \\
 &= L \cdot R^0 \cdot R^1 \\
 &= \begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \dots & \dots & \dots & \\ l_{n,1} & l_{n,2} & \dots & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} r_{1,1} & & & \\ & r_{2,2} & & \\ & & \ddots & \\ & & & r_{n,n} \end{bmatrix} \cdot \begin{bmatrix} 1 & \frac{r_{1,2}}{r_{1,1}} & \dots & \frac{r_{1,n}}{r_{1,1}} \\ & 1 & \dots & \frac{r_{2,n}}{r_{2,2}} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix} \right)
 \end{aligned} \quad (2)$$

(C) The KGC saves all the information of the matrix R , and according to the diagonal elements of the matrix R , the following one-way polynomial function is used to generate the identification number Id_i for each CH node i .

$$\begin{aligned}
 Id_i &= f(x) \\
 &= (r_{11} \cdot x^{(n)} + r_{22} \cdot x^{(n-1)} + \dots + r_{nn} \cdot x^{(1)}) \bmod (p), \quad (3)
 \end{aligned}$$

$$\text{where } x = r_{ii}, i \in (1, n).$$

(D) KGC broadcasts the matrix L to all the CHs; that is to say, each CH knows all the information of the matrix L .

(E) According to the CH number Id_i and matrix L , each CH node only stores the corresponding row elements in

matrix L and ignores other elements. For example, if $Id_i=4$, the CH only stores the 4th row elements of the L matrix as follows:

$$\begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \dots & \dots & \dots & \\ l_{n,1} & l_{n,2} & \dots & 1 \end{bmatrix} \xrightarrow[i=4]{Id_i} \begin{bmatrix} \dots & \dots & \dots & \\ \dots & \dots & \dots & \\ \dots & \dots & \dots & \\ l_{4,1} & l_{4,2} & \dots & 1 \end{bmatrix}. \quad (4)$$

3.2.2. User Key Generation (Extract). After initialization, the steps of key generation are described as follows.

(A) The applicant node i , named as CH_i , selects a point $P(x_i, y_i)$ on the elliptic curve as its own private key, that is, $S_i=(x_i, y_i)$, and then randomly selects a number $z_i \in \mathbb{Z}$ and calculates the verification shares $V_i=z_i S_i=(V_{xi}, V_{yi})$.

(B) CH_i saves the random number z_i , which will be used in the signcryption process; then it combines the self verification shares V_i with the identity information Id_i and sends it to the KGC. It should be noted that random number z_i will not be sent to KGC, which prevents KGC from leaking secrets, and z_i is only used for the intercluster communication.

(C) After receiving the information, the KGC selects a random number j . According to the value of j together with CH_i 's number i , the KGC chooses the element r_{ji} in the matrix R and obtains the verification parameter $V_{Kgc} = (r_{ji}/r_{jj})$ of the KGC.

(D) The KGC calculates the public key $P_i = G \cdot V_i$ for CH_i based on the received verification shares $V_i(V_{xi}, V_{yi})$.

(E) The KGC sends the random number j , the verification parameter V_{Kgc} , and the user public key P_i to the applicant CH_i .

(F) After receiving the information returned by the KGC, according to the received random number j , CH_i selects the j th element l_{ij} from the stored i th row vector in matrix L and verifies whether $l_{ij} = V_{Kgc}$. If the verification passes, CH_i accepts the public key P_i and saves its own private key S_i . Then, the KGC announces the user's public key P_i to other CHs. Otherwise, it returns an error and the application fails (the correctness will be proven in Section 4.1).

Thus, the configuration of the public and private keys for each CH has been completed.

3.2.3. Signcryption. When CH_i needs to send a message to CH_j , the following steps will be performed.

(A) CH_i first takes the random number z_i which is saved at the public key application stage; together with the base point G on the elliptic curve, CH_i calculates $K_i = z_i \cdot G = (k_x^{Gi}, k_y^{Gi})$. Then, CH_i calculates the verification parameter $U_i = k_x^{Gi} \cdot P_j = (u_x^{ij}, u_y^{ij})$ by k_x^{Gi} and the public key P_j of CH_j . After that, CH_i encrypts the message Msg by using u_x^{ij} and obtains $C = E_{u_x^{ij}}(Msg)$.

(B) CH_i calculates the parameters $M_i = H1(k_x^{Gi} \parallel C)$, $N_i = H0(k_x^{Gi} \cdot M_i)G$,

$$O' = Si - (k_x^{Gi} \cdot M_i) \bmod q, \quad (5)$$

$$O = M + O'$$

(C) CH_i generates the ciphertext $\sigma_i = (K(k_x^{Gi}, k_y^{Gi}) \parallel C = E_{u_x^{ij}}(Msg) \parallel N_i \parallel O \parallel z_i)$ according to the stored random number z_i and all the above parameters.

(D) CH_i sends the ciphertext to CH_j .

3.2.4. Unsigncryption. After CH_j receives the ciphertext, the following steps will be performed to decrypt the received message.

(A) CH_j extracts the first half k_x^{Gi} of K_i and calculates the verification parameters $U_j = k_x^{Gi} \cdot P_j = (u_x^j, u_y^j)$ with its own public key.

(B) CH_j calculates

$$M_j = H1(k_x^{Gi} \parallel C) \quad (6)$$

$$O'' = O - M_j.$$

(C) CH_j decrypts the encrypted message by using u_x^j to obtain $Msg = D_{u_x^j}(C)$.

(D) CH_j calculates $P_j' = N_i + O'' \cdot G$ and verifies if $z_i \cdot P_j = P_j'$; if they are equal, the common communication key of CH_i and CH_j is u_x^{ij} , and u_y^{ij} is stored as the intercluster shared key after receiving the text Msg ; otherwise, the verification fails.

Thus, communication between the clusters finished.

3.3. Signcryption for Intracluster Communication

3.3.1. Key generation and Unsigncryption. In each cluster, CH_i and CH_j obtained the shared key u_y^{ij} between cluster i and cluster j through the above process. The key generation, signcryption, and unsigncryption for the ordinary nodes in a cluster are similar to those for cluster head, as described in Section 3.2, so it is not necessary to repeat them here.

3.3.2. Intracluster Key Updating. In order to ensure security, the BS will periodically inform each CH to perform the intracluster key updating operation. For instance, in the cluster whose cluster head is CH_i , the updating steps are as follows:

(A) CH_i generates a new cluster shared key $(u_y^{ij})'$.

(B) CH_i uses its own private key S_i for identity signature, and the ordinary nodes authenticate it.

(C) The signature and the new cluster shared key $(u_y^{ij})'$ together with the original shared key are encrypted to obtain $C' = E_{u_x^{ij}}(S_n(Id_i)S_i \parallel Id_i \parallel Clock \parallel (u_x^{ij})')$ and the information is broadcasted in this cluster.

(D) After the ordinary nodes receive the broadcasted information, they decrypt the information by original shared key u_y^{ij} and verify it by the public key of the cluster head; if the verification passes, the new cluster shared key $(u_y^{ij})'$ is saved to replace the old one and answers CH ; otherwise, discard the data package.

4. Correctness and Security Analysis

4.1. Correctness Analysis

Theorem 1. In the process of generating public key (described in Section 3.2.2) for CH , if $l_{ij} = V_{KGC}$, the public key generation satisfies the correctness requirements.

Proof. Firstly, according to the Doolittle decomposition rules for the symmetric matrix D , combined with the knowledge of Order Principal Minor Determinant, we have

$$D = L \cdot R = \begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \dots & \dots & 1 & \\ l_{n,1} & l_{n,2} & l_{n,n-1} & 1 \end{bmatrix} \cdot \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ & r_{2,2} & \dots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{bmatrix} \quad (7)$$

because $|D| = |L \cdot R| = |L| \cdot |R|$, $|D| \neq 0$, $|L| = 1$

that is $|D| = 1 \cdot |R| = \prod_{i=1}^n r_{ii} \neq 0 \implies$

$r_{ii} \neq 0 (i \in 1, 2, \dots, n)$

Furthermore, the matrix R can be decomposed as follows:

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,n} \\ & r_{2,2} & \dots & r_{2,n} \\ & & \ddots & \vdots \\ & & & r_{n,n} \end{bmatrix} = R^0 \cdot R^1 \quad (8)$$

$$= \begin{bmatrix} r_{1,1} & & & \\ & r_{2,2} & & \\ & & \ddots & \\ & & & r_{n,n} \end{bmatrix} \cdot \begin{bmatrix} 1 & \frac{r_{1,2}}{r_{1,1}} & \dots & \frac{r_{1,n}}{r_{1,1}} \\ & 1 & \dots & \frac{r_{2,n}}{r_{2,2}} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}$$

TABLE 2: Function $y=f(x)$.

Factor	r_{11}	r_{22}	r_{33}	...	$r_{(n-1)(n-1)}$	r_{nn}
Y/x_i	$Y_1=r_{11}$	$Y_2=x_i \cdot Y_1 + r_{22}$	$Y_3=x_i \cdot Y_2 + r_{33}$	...	$Y_{n-1}=x_i \cdot Y_{n-2} + r_{(n-1)(n-1)}$	$Y_n=x_i \cdot Y_{n-1} + r_{nn}$

Therefore, we have

$$D = L \cdot R = L \cdot R^0 \cdot R^1$$

because $D = D^T$, $D = L \cdot R^0 \cdot R^1$

$$\begin{aligned} \text{so } D &= D^T = (L \cdot R^0 \cdot R^1)^T = (R^0 \cdot R^1)^T \cdot L^T \\ &= (R^1)^T \cdot (R^0)^T \cdot L^T \\ &= (R^1)^T \cdot R^0 \cdot L^T \\ &= L \cdot R^0 \cdot R^1 \end{aligned} \quad (9)$$

Based on uniqueness of the Doolittle decomposition, we have $L = (R^1)^T$; that is,

$$\begin{aligned} L &= \begin{bmatrix} 1 & & & \\ l_{2,1} & 1 & & \\ \dots & \dots & \dots & \\ l_{n-1,1} & l_{n-1,2} & \dots & 1 \end{bmatrix} = R^T \\ &= \begin{bmatrix} 1 & \frac{r_{1,2}}{r_{1,1}} & \dots & \frac{r_{1,n}}{r_{1,1}} \\ r_{2,1} & 1 & \dots & \frac{r_{2,n}}{r_{2,2}} \\ & & \ddots & \vdots \\ & & & 1 \end{bmatrix}^T \\ V_{Kgc} &= \begin{pmatrix} \frac{r_{ji}}{r_{jj}} \\ \vdots \end{pmatrix} \\ &\Downarrow \\ l_{ij} &= V_{Kgc} \end{aligned} \quad (10)$$

Therefore, the KGC public key generation process satisfies the correctness requirements. $\square$

Theorem 2. *The recalculation process of the verification parameter P_i satisfies the correctness requirements.*

Proof.

$$\begin{aligned} \text{since } O' &= S_i - H_0(k_x^{Gi} \cdot M_i) \bmod q, \\ N_i &= H_0(k_x^{Gi} \cdot M_i) G, \\ O'' &= O - M_i \\ \text{then } P_j' &= N_i + O'' \cdot G \\ &= H_0(k_x^{Gi} \cdot M_i) G + (O - M_i) G \\ &= H_0(k_x^{Gi}) G + M_i \cdot G + O \cdot G - M_i \cdot G \end{aligned}$$

because $P_i = V_i \cdot G$,

$$V_i = z_i \cdot S_i$$

$$\text{so } P_i = (z_i \cdot S_i) \cdot G = z_i (S_i \cdot G) = z_i \cdot P_j' \quad (11)$$

$\square$

4.2. Security Analysis. In the HMDS algorithm, the private key of an applicant is managed by the applicant itself, and the KGC only generates the public key for the applicant. The authentication parameters transferred during the communication process are also based on the public keys of the applicant. Therefore, even if a node is compromised, it will not affect the security of other nodes.

In the HMDS algorithm, one characteristic is that the monomial function is adopted to generate the identification number for an applicant, which ensures the partial public secret not to leak. From the polynomial, $Id_i = f(x) = (r_{11} \cdot x^{(n)} + r_{22} \cdot x^{(n-1)} + \dots + r_{nn} \cdot x^{(1)}) \bmod (p)$; according to Horner's Rule, we can continuously extract x_i as a common factor from the residual polynomial recursively to compute Y_i , as illustrated in Table 2. It can be observed that if x , r_{ii} , and p are known, it is very easy to compute the value of Y_i . Conversely, if Y_i and p are known and we try to get the value of x_i , it will cost at least $n^2(\log_2 P)^2$ times of multiplication. When n and p are very large, it is very difficult to obtain the value of x_i . Therefore, it is ensured that the identification number of each node is quickly generated, but it is difficult to crack. Another characteristic of the HMDS algorithm is that the information kept by each applicant is very simple, which saves the cost of calculation and storage resource.

5. Performance Evaluation

5.1. Computation Complexity Analysis. We compare the HMDS algorithm with the HIS algorithm; the main operations and complexity of the two algorithms are shown in Table 3, which include four steps, system initialization, user key generation (extraction), signcryption, and unsigncryption.

5.2. Simulation Experiments. We set up six classical simulation configurations for HMDS and HIS scheme to represent different scenarios, denoted as S1~S6. To evaluate the efficiency of the HMDS scheme, we take the time cost as the metric to compare the HMDS scheme with the HIS scheme. The experiment parameters setting of the 6 scenarios is as shown in Table 4, where Num_CH is the number of clusters and n is the number of nodes.

Since the key steps of the HMDS scheme are setup (initialization) stage and key generation (extract) stage, the efficiency of the HMDS scheme depends on the two stages.

TABLE 3: Complexity of calculation.

Process	HMDS	HIS
	Operation	Operation
	cost of calculation	cost of calculation
Initialization	(A) Select D;	(A) Select q, Gq, E(G);
	(B) Doolittle decomposition :	(B) Select Symmatrix D;
	(C) $r_{ii} \xrightarrow{f(x)} Id_i$;	(C) Genetate H0, H1;
	(D) $Matrix[L] \rightarrow Id_i$;	(D) Vandermonde Matrix ID;
	(E) $Id_i \xleftarrow{Saved} Row_i(L)$	(E) Generate key;
Extraction	(A) Select Pi; Generate Vi;	(F) Broadcast;
	(B) $Id_i \xrightarrow{Vi Id_i} KGC$;	(A) Select Pi; Generate Vi;
	(C) Calculate VKgc;	(B) $Id_i \xrightarrow{Vidi} KGC$;
	(D) Calculate Pi;	(C) $D \xrightarrow{y=f(x) \bmod(p)} D'$;
	(E) $KGC \xrightarrow{j V_{kg} P_i} Id_i$;	(D) Caculate $(D' * ID) :$
	(F) $l_{ij} = V_{Kgc}$	(E) $KGC \xrightarrow{K_{ij} B_i P_i} Id_i$;
	(A) Generate Ki, Ui, C;	(F) Caculate $(B * ID) :$
Signcryption	$\left\{ \begin{array}{l} M_i = H_1(k_x^{Gi} C) \\ N_i = H_0(k_x^{Gi} \cdot M_i)G \\ O' = S_i - H_0(k_x^{Gi} \cdot M_i) \bmod q \\ O = M + O' \end{array} \right.$	(A) IDi : Generate K, U, C = Eu(m);
		(B) Calculate M, N;
		(C) Calculate O', O = M + O' ;
	(C) $Id_i \xrightarrow{\sigma_i=(K(k_x^{Gi}, k_p^{Gj}) C)} Id_j$;	(D) $Id_i \xrightarrow{\sigma=(K(k_1, k_2), G, N, O, x)} Id_j$;
Unsigncryption	(A) Calculate Uj;	
	$\left\{ \begin{array}{l} M_j = H_1(k_x^{Gj} C) \\ O'' = O - M_j \end{array} \right.$	(A) Calculate $U' = k_1 P_j = (u, v)$;
	(C) $Msg = D_{u_x}^{Gj}(C)$	(B) $M = H1(k1 C), O' O - M$;
	$\left\{ \begin{array}{l} P'_j = N_i + O'' \cdot G \\ z_i \cdot P'_j \equiv P_i \end{array} \right.$	(C) $m = D_u(C)$;
Total		(D) $\left\{ \begin{array}{l} P'_j = N_i + O' \cdot G \\ z_i \cdot P'_j \equiv P_i \end{array} \right.$
	O(HMDS)=n*n	O(HIS)=6n*n

TABLE 4: Experimental setting.

OS	Cpu	HZ	RAM	Developing environment	Library files	Initial energy of ch (Joule)	Energy range	Number of cluster layers
Win10 home	i5,x64	1.9Ghz	8G	Visual studio 2010	<time.h>clock()	20	1.5-3J (random)	2
		S1:n=25,Num_CH=2; S4:n=300,Num_CH=10;		S2:n=60,Num_CH=3; S5:n=600,Num_CH=20;		S3:n=100,Num_CH=4; S6:n=1000,Num_CH=35;

FIGURE 2: Time consumption of the two stages.

Therefore, besides evaluating the overall efficiency of the HMDS scheme, we specially evaluate the efficiency of the two stages, respectively.

5.2.1. Efficiency of Initialization and Extract Stage. As shown in Figure 2, it can be observed that the time consumption of the two algorithms in the three scenarios, S1, S2, and S3, is all within 100ms and the HMDS algorithm has a slight advantage over the HIS algorithm. It is because the number of the nodes is small. With the number of nodes increasing from 100 (in S3) to 1000 (in S6), the time consumption of the two algorithms also has a large increment, and the HMDS algorithm has an obvious advantage over the HIS algorithm. Especially in the scenario S6, the HIS algorithm takes about 850ms, while the HMDS algorithm only takes 300ms, which saves about two-thirds of the time. In general, for the HIS algorithm, the time consumption of the two stages increases steeply when the network size n increases quickly. In comparison, the HMDS algorithm is relatively stable in time consuming when the network size n is large. Therefore, the HMDS algorithm is more applicable to the WSN than the HIS algorithm. Especially when the network size n is very large, the HMDS algorithm has more obvious advantages.

5.2.2. Overall Efficiency. We randomly take 20 pairs of nodes to evaluate the overall efficiency; that is to say, we measure the time consumption of the four stages, from initialization to the end of unsigncryption. As shown in Figure 3, the overall time consumption for both algorithms increases with the number of nodes increasing, but the HMDS algorithm still takes less time than the HIS algorithm. This trend becomes more and more obvious with the increasing of the network size and is consistent with the results of the above two-stage experiments.

Therefore, the HMDS algorithm outperforms the HIS algorithm. Furthermore, the HMDS algorithm does not cause rapid increase of time consumption when the network scale grows. So the HMDS algorithm is more suitable for larger-scale WSN.

6. Conclusion

In this paper, the identity-based HMDS authentication algorithm was proposed. The public key of the applicant is generated by KGC and the private key is computed by the node

FIGURE 3: Overall time consumption.

itself to solve the key escrow problem, which is a disadvantage of the common identity-based authentication algorithm. The HMDS algorithm adopts hierarchical structure to adapt to WSN requirement. By employing matrix decomposition, the cost of calculation and storage is reduced. Through analysis and experimental comparison, the HMDS algorithm can ensure communication security in the WSN environment and has characteristics of low energy consumption and high stability, which is very suitable for WSN. Furthermore, when the network scale is large, the performance of the HMDS algorithm is very stable and has advantages over HIS scheme. In the future, we will improve the existing matrix decomposition-based HMDS algorithm, especially to optimize the authentication method between cluster heads. Meanwhile, we will study the key updating mechanism to improve the authentication efficiency.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is partly supported by National Natural Science Foundation of China (Grant no. 11671400).

References

- [1] M. Guo, Y. Liu, H. Yu, B. Hu, and Z. Sang, "An overview of smart city in China," *China Communications*, vol. 13, no. 5, pp. 203–211, 2016.
- [2] H.-D. Ma, "Internet of things: objectives and scientific challenges," *Journal of Computer Science and Technology*, vol. 26, no. 6, pp. 919–924, 2011.
- [3] W. Chen, X. Jiang, Z. Tang et al., "Context-based global multi-class semantic image segmentation by wireless multimedia sensor networks," *Artificial Intelligence Review*, vol. 43, no. 4, pp. 579–591, 2015.

- [4] H. Y. W. L. and P. Z., "Intelligent wireless sensor network system," *Science Publishing Company*, 978-7-03-037323 -6, 2013.
- [5] K. T. Tran and S.-H. Oh, "UWSNs: A Round-Based Clustering Scheme for Data Redundancy Resolve," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 383912, pp. 19–26, 2014.
- [6] Y. Z. Song and L. W. Li, "Dynamic triggering and cooperation based lifespan optimization of WSN nodes," *Control Engineering of China*, vol. 20, no. 2, pp. 339–343, 2013.
- [7] H. J. Cheng, X. B. Huang, and N. Xiong, "Minimum-Energy broadcast algorithm for wireless sensor networks with unreliable communications," *Journal of Software*, vol. 25, no. 5, pp. 1101–1112, 2014 (Chinese).
- [8] P. Khandare and N. P. Kulkarni, "Public key encryption and 2Ack based approach to defend wormhole attack," *International Journal of Computer Trends & Technology*, vol. 4, no. 3, pp. 421–428.
- [9] Y. L. Zhang, D. R. Zhou, and C. Y. Li, "Certificateless-based efficient aggregate signature scheme with universal designated verifier," *Journal of Communications*, vol. 36, no. 2, pp. 1–5, 2015.
- [10] [http://en.wikipedia.org/wiki/RSA_\(algorithm\)](http://en.wikipedia.org/wiki/RSA_(algorithm)).
- [11] J. Huang and B. Huang, "Public key based key distribution scheme for wireless sensor networks," *Tongxin Xuebao/Journal on Communication*, vol. 32, no. 10, pp. 52–58, 2011.
- [12] Z. G. Qin, J. J. Liu, Y. Zhao et al., "A survey of key-insulated cryptography," *Chinese Journal of Computers. Jisuanji Xuebao*, vol. 38, no. 4, pp. 759–774, 2015.
- [13] X. Chen, S. Y. Liu, and Y. Wang, "Emergency resources scheduling based on improved backtracking search optimization algorithm," *Computer Applications and Software*, vol. 32, no. 12, pp. 235–238, 2015.
- [14] L. B. Oliveira, R. Dahab, J. Lopez, F. Daguano, and A. A. Loureiro, "Identity-Based Encryption for Sensor Networks," in *Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07)*, pp. 290–294, White Plains, NY, USA, March 2007.
- [15] C. Yuan, W. Chen, and D. Li, "A Hierarchical Identity-Based Signcryption Scheme in Underwater Wireless Sensor Network," in *11th China Conference on Wireless Sensor Network (CWSN 2017)*, 2017.
- [16] J. Guo, X. Li, and J. Wu, "New Identity-based key agreement scheme for WSN," *Computer Science*, vol. 38, no. 3, pp. 127–130, 2011.
- [17] J. Katz and M. Yung, "Scalable protocols for authenticated group key exchange," *Journal of Cryptology*, vol. 20, no. 1, pp. 85–113, 2007.
- [18] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO 84 on Advances in cryptography*, vol. 196, LNCS 196, pp. 47–53. New York, NY, USA, Springer-Verlag New York, Inc., 1985.
- [19] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *Proceedings of the 16th IEEE Computer Security Foundations Workshop, CSFW 2003*, pp. 219–233, USA, July 2003.
- [20] J.-M. Bohli, B. Glas, and R. Steinwandt, "Towards provably secure group key agreement building on group theory," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 4341, pp. 322–336, 2006.
- [21] F. Zhang, S. Liu, and K. Kim, "ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings," *Cryptology ePrint Archive*, Report 2002/122, 2002.
- [22] N. Kobitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.
- [23] V. S. Miller, "Use of Elliptic Curves in Cryptography," in *Lecture Notes in Computer Science*, vol. 218, pp. 417–426, 1 edition, 1985.
- [24] W.-F. Zhang, X.-M. Wang, W. Guo, and D.-K. He, "An efficient inter-enterprise authentication scheme for VE based on the elliptic curve cryptosystem," *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, vol. 42, no. 6, pp. 1095–1102, 2014.
- [25] P. Guo, H. Zhang, D. Fu, and M. Zhou, "Hybrid and Lightweight Cryptography for WSN," *Computer Science*, vol. 39, 2012.
- [26] D. W. Zhou, G. H. Wei, and H. G. Zhang, "Key-management scheme based on public-key institution to clustered wireless sensor networks," *Journal of Beijing University of Technology. Beijing Gongye Daxue Xuebao*, vol. 42, no. 5, pp. 707–712, 2016.
- [27] G. Lippold, C. Boyd, and J. Gonzalez Nieto, "Strongly secure certificateless key agreement," in *Pairing-based cryptography—Pairing 2009*, vol. 5671 of *Lecture Notes in Comput. Sci.*, pp. 206–230, Springer, Berlin, 2009.
- [28] R. S. Zhang and K. S. Qu, "Energy hole alleviation of WSNs based on dual cluster head grid scheduling," *Transducer and Micro-system Technologies*, vol. 33, no. 10, pp. 133–136, 2014.
- [29] Z. Dai, C. Yan, and Z. Wu, "New uneven double cluster head clustering algorithm for WSN-PUDCH algorithm," *Chinese Journal of Sensors and Actuators*, vol. 29, no. 12, pp. 1912–1918, 2016.
- [30] L. Wang, "Improved algorithm of non-uniform clustering routing protocol[J]," *Computer Science*, vol. 44, no. 2, pp. 152–156, 2017.
- [31] X. Liu, "A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 67, pp. 43–52, 2016.
- [32] M. Simões, L. C. Simões, S. Cleto, I. Machado, M. O. Pereira, and M. J. Vieira, "Antimicrobial mechanisms of orthophthalaldehyde action," *Journal of Basic Microbiology*, vol. 47, no. 3, pp. 230–242, 2007.
- [33] R. Linsker, "Self-organization in a perceptual network," *The Computer Journal*, vol. 21, no. 3, pp. 105–117, 1988.
- [34] S. Deng and Y. Wang, S. Deng, vol. 39, Chinese, *Computer Engineering*, 2013.
- [35] E. Klaoudatou, E. Konstantinou, G. Kambourakis, and S. Gritzalis, "A survey on cluster-based group key agreement protocols for WSNs," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 3, pp. 429–442, 2011.