Telecommunication Systems (2022) 79:3–16
https://doi.org/10.1007/s11235-021-00844-4

# An efficient secure key establishment method in cluster-based sensor network

**Akansha Singh**[1] · **Khushboo Jain**[2]

Accepted: 29 September 2021 / Published online: 18 October 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

**Abstract**
The main issue for the security of wireless sensor networks (WSNs) is how to allow sensor nodes (SNs) to establish and share cryptographic keys in an energy-efficient, storage-efficient, and authentic manner for their secure data transmission. Furthermost recent studies carried out in this direction is concerned with homogeneous networks in which all sensor has identical characteristics and fundamental administration mechanisms. However, Cluster-based sensor networks have demonstrated better achievements and performance than homogeneous networks because of the several benefits of clustering. This inspired us to propose a secure key-establishment method for cluster-based sensor networks based on symmetric-key cryptography. Since symmetric key cryptography has small energy consumption, they are a great choice to prefer for securing the networks. Even though symmetric key cryptography has high storage needs, this deficiency can be reduced by using suitable methods—evaluating the proposed work that the storage needs are reduced along with reduced energy consumption. The work offers a favorable level of security against various intruders and possible security threats and is additionally scalable than the state-of-the-art techniques.

**Keywords** Cluster-based sensor networks · Communication cost · Key storage · Hash function · Secure key establishment method · Wireless sensor networks

## 1 Introduction

Wireless Sensor Networks (WSNs) have integrated several application domains, including military and non-military areas, has now become a significant part of our lives. WSNs usually comprise a set of low-cost and resource constraint sensor nodes (SNs) for sensing and data collecting over a specific application domain. Since the WSNs are infrastructure less and randomly deployed, the SNs usually suffer from several challenges like energy management, storage, and communication and computation abilities [1, 2].

Due to these limitations and challenges of WSNs, we inculcate the standard techniques with them. For instance, most cryptographic algorithms such as RSA, Diffie-Hillmen, ECC, DSA [3–6] are based on asymmetric key encryption, also known as (Public-key cryptography) that incorporates two separate keys. The key distribution challenge is eradicated because there is no need to exchange the secret keys; therefore, the security is strengthened as the keys do not have to be communicated to anybody.

Despite these benefits, they are not a promising solution for in WSNs due to the high energy consumptions. Therefore, the cryptographic techniques should be selected, which suits the environments and handles the challenges and constraints of these networks. One way to categorize WSN is as homogeneous and heterogeneous systems. In homogeneous WSNs, each SNs have similar roles and capabilities regarding energy, storage, and computation power. However, homogeneous sensor networks have their constraints and challenges, which are stated in some studies [3, 4].

Since the data communication between SNs is significant, particularly in military applications, and is transmitted through a wireless channel, a mechanism is needed to ensure the security, integrity, and confidentiality of sensed transmissions. One of the primary tasks of symmetric key cryptography-based key management protocols is to gen-

✉ Khushboo Jain
khushboojain2806@gmail.com

Akansha Singh
singhakansha1@gmail.com

[1] Noida Institute of Engineering & Technology, Greater Noida, India

[2] DIT University, Dehradun, India

erate mutual secret keys for the SNs so that they can use essential security facilities, like secrecy of communications, the integrity of data, and authentication by means of these keys.

Both symmetric-key algorithms (also known as private key cryptography) and asymmetric key algorithms (also known as public-key cryptography) are used to secure sensed data in WSNs. Symmetric procedures such as RC4 AES, DES [6–8] have less energy consumption and are fast, but their storage need is undoubtedly high. Though inculcating appropriate methods, which is presented in the following work, we can handle these challenges.

Therefore, it is feasible to use general features of the private key cryptosystem, which are faster processing, lesser energy ingesting, and higher storage consumption resulted from storing the keys. Whereas the asymmetric key cryptosystem has low storage, but they consume high energy in comparison to symmetric algorithms [5]. This problem roots private key algorithms to be more favorable for use in sensor networks because the SNs are not able to run asymmetric algorithms for an extended period of time because they have insufficient energy of SNs.

We can secure the sensor's data in this work by encrypting the sensed data before communicating with only a single key for Cluster Head (CH) and dual keys for every SN; this technique is used in sensor systems with network size. Therefore, the number of keys stored in different SNs will not be dependent on the number of CHs and SNs since they will be constant in all scenarios. In heterogeneous WSNs, also known as cluster-based sensor networks, the SNs are divided into three categories; SNs with constraint resources can sense, collect and forward data to the nodes with higher resources known as CHs; CH then aggregates and forwards this data to the Base-station (BS) which has the highest number of resources in the network. SNs in each cluster directly communicate to their respective CHs or via other SNs which belong to that cluster. CHs of each cluster can communicate to BS either directly or indirectly via other CHs of the network [9–11]. The sample architecture of such cluster-based network architecture is shown in Fig. 1.

To ensure that all sensed and transmitted data is secure, the following requirements should apply:

1. If an SN is directly linked to a CH without an intermediate SN, a key must be exchanged between them.
2. If an SN is linked to a CH via another SN of that cluster, a key must be exchanged between those SNs.
3. If a CH is linked directly to the BS, a key must be exchanged between them.
4. If a CH is linked to the BS indirectly, i.e., via another CH, a key must be exchanged between them.

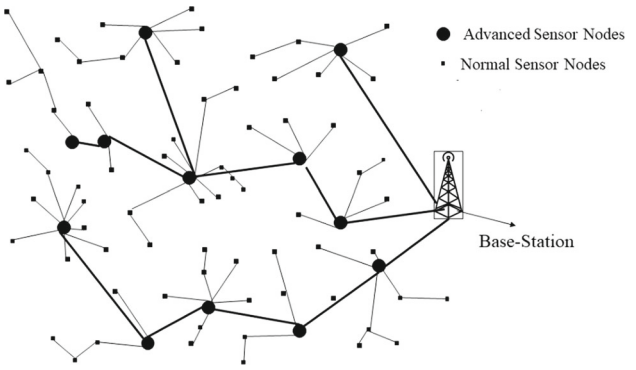


**Fig. 1** A sample Cluster-based Sensor Network Architecture

If these above preconditions can be fulfilled by a key establishment process, we can be confident that all collected data in SNs will be retrieved in an encrypted form by BS, and the attackers will not be able to access the sensed data.

In our proposed work, we consider the BS as the only competent authority to authenticate admission of new SNs and generation of keys. Initially, it may appear like the BS will be a performance barrier for the sensor network under certain conditions, and the entire network architecture can be ruined after dropping the BS, but it is totally incorrect because the BS will be a trustworthy entity and rich resource. Even like other SNs, it is not positioned in unattended areas.

### 1.1 Contribution of the work

The main contributions of this paper in this direction are:

1. To propose a secure key establishment method (SKEM) technique based on symmetric-key cryptography since they are faster than asymmetric and have less energy consumption fast, but the storage consumption is certainly high. Though, inculcating appropriate methods, which is presented in the following work, can counteract this drawback.
2. The proposed SKEM technique can secure all sensor data by encrypting the sensed data before communicating with only one key in each CH and two keys in each SN.
3. For the purpose of performance evaluation of the SKEM technique is evaluated for the issues of energy in terms of computation and communication; storage needs in terms of key storage, scalability, and Complexity, which is helpful in the effectiveness assessment of the proposed scheme.

### 1.2 Organization of the work

The rest of the paper is systematized as follow: Chapter 2 discusses the background related work associated with the proposed scheme, chapter 3 throws light on network archi-

tecture and annotation, which laid the foundation of the proposed work, chapter 4 gives the description of the working of the Secure key establishment method (SKEM), chapter 5 presents the security investigation and discusses how the SKEM is able to forbid few potential security threats and attacks. Chapter 6 presents the simulation with state of art and discusses the results, and chapter 7 gives the conclusion of the work.

## 2 Related work

This section gives a brief description of the previous methods presented for secure key establishment in WSNs.

Mizanur and El-Khatib [12] presented a key-agreement procedure that is based on key pair cryptography over an elliptic curve. This algorithm is called RE; by making use of the pairing and identity-based encryption characteristics, any two SNs that need to communicate will compute the same private key independently. The suggested protocol significantly reduces an SN's key-space and has been resilient against a variety of attacks. Later, several networks used this model.

Shi et al. [13] proposed a lightweight technique for HWSNs to balance the sensor's resources called RAKE, which enables SNs to create shared keys in an authentic and resource-efficient way for their secure data transmissions. This provides a unique way of establishing vertical-key shareability before exchange for allowing horizontal-key shareability for authentic shared key establishment after deployment. The assessment of the method demonstrates that it provides fault tolerance and high authenticity to different security attacks and is additionally resource-efficient, versatile, and more scalable than state-of-the-art.

Khemissa et al. proposed an Ultra-lightweight authentication method [14] for the HWSN scheme for both the SNs as well as the application or user to authenticate each other to establish secure data transmission. To achieve mutual authentication, this work uses only nonce, exclusive-or, concatenation procedures. Furthermore, it terminates with a session key agreement between the SN and the user. The efficiency and security review highlights that authentication with low energy consumption is provided by the proposed work and ensures resistance to various types of attacks.

Darbandehet al. [15] suggested a security protocol to enhance the work of Khemissaet al. [14] as a protocol in the view of desynchronization, user impersonation, and gateway impersonation attacks are not adequately safe. The improved protocol offers an acceptable level of protection and also uses two formal ways to prove its security, i.e., BAN and even the Scyther tool.

A two-factor authentication method for WSNs was developed by Jiang et al. [16], which is based on ECC, and the practical and security faults of this method are analyzed. Based on this. Li et al. [17] proposed a three-factor authentication method to improve the security flaws of the [16] scheme for IoT-based WSNs, where the fuzzy commitment method was implemented for improving the processing efficiency and also attains enhanced security.

An extremely protected CAKE method is proposed by Mehra et al. [18], which is based on one-way hashing with codeword authentication and a one-time password. For the analytical estimation of the method, the Random Oracle Model and BAN are used, and the AVISPA method is used to simulate it. Confidentiality, integrity, and authentication are maintained by the proposed protocol and can counter many attacks such as offline guessing attacks, replay attacks, impersonation attacks, etc., and retain perfect forward secrecy.

Athmani et al. [19] proposed an efficient, dynamic authentication and key Management technique for HWSN. The main concept is to provide both authentication and key establishment with a single lightweight protocol while maximizing the degree of protection. In order to generate dynamic keys, the key distribution algorithm is based on pre-existing data and does not need a secure channel and sharing process that enhances protection, energy efficacy and decreases memory usage.

Mirvaziri et al. [20] proposed a key establishment method based on symmetric cryptography as it consumes to have low energy as compared to the asymmetric one. While high memory consumes symmetric cryptosystems, this vulnerability can be minimized by suitable techniques. The results of the simulation showed that the proposed work would substantially reduce the memory consumption resulting from the saving of keys as well as reducing the energy consumption resulting from communications in all network nodes compared to similar works.

With the aid of the ECDSA cryptographic scheme, Qazi et al. [21] proposed a scheme that provides protection for the node-to-node communication network but also reduces memory consumption to provide an effective mechanism for measuring key generation time numbers of hello messages and packet size. In addition, key management with appropriate key duration is also given by the ASCW. In addition, ASCW helps to protect node-level communication, which helps to secure the entire network in a safer and more effective way. With the aid of the authentication system, ASCW also lowers the cost of risk and security threats on the network.

Khashan et al. [22] proposed a lightweight encryption scheme for secure and energy-efficient to overcome the issues of key management, authentication, flexibility, and resource management. The FlexCrypt system is a dynamic clustering mechanism that supports mobility among the SNs is designed to overcome these challenges. By automating the selection of

encryption settings, this system gives a flexible, lightweight cryptographic strategy for controlling the Complexity of the encryption. The key management and authentication mechanism is also intended to ensure secure communication and data and key exchange across the various SNs. According to Singh et al. [23], an attacker can compute the secret parameter that is used by a gateway during connection with others. Now the attacker has the ability to change the first message sent by the user to the sensor node. They presented an improved approach in order to address the shortcomings.

## 3 Key establishment method based on symmetric-key cryptography

This section presents the network architecture and the preliminaries for the proposed key establishment method, which is established on Symmetric-key Cryptosystem, and the abbreviation and annotation, which is used by the proposed SKEM technique.

### 3.1 Network architecture and set-up

This paper considers a Key Establishment Method, which is based on a cluster-based layered sensor network architecture, as shown in Fig. 1. The first layer consists of the SNs, equipped with limited resources. SNs continuously forward the sensed data to the respective CH, which is also battery operated but is better equipped with resources. Data Aggregation will be done at the CHs to reduce the number of data packets. Clustering-based schemes are promising techniques [24] as they offer good scalability and support for data aggregation in WSNs. The communication within a cluster is managed by the CH, which is on the second layer of the architecture. The third layer consists of one BS, which advances from rich communication and computation resources and does not require a battery. BS allows all the CHs to aggregate and forward the inter-cluster information. All SNs and CHs are uniformly and randomly deployed in a 2D space and assigned a unique ID.

- SN at the first tier is homogeneous in the normal sensor nodes (NSN) and is not armed with Tamperproof hardware.
- Each NSN can belong to only one cluster for every communication round.
- NSN has a low communication range and can intercept only the transmission of their nearby neighbor.
- CHs at the second tier are the advanced sensor nodes (ASN) are equipped with Tamperproof hardware.
- Each ASN has a high communication range and can listen to other ASN.
- All NSNs and ASNs will have unique IDs.
- NSNs can only transmit their sensed data to the BS via ASNs only.
- BS at the third tier has no energy constraints and is the trusted entity, i.e., it cannot be compromised by an attacker.
- BS can listen and communicate with all ASN in the sensor network.

The CH may interact with all NSNs directly in each cluster, but an NSN may require one or more hops to interact with its CH. With the support of neighbor CHs, CH, which is farther away from the BS, can connect with the BS via hop-by-hop. Each NSN also records other ASNs from which Hello messages have been sent at the same time, and these ASNs will act as backup CHs in the event that the CH fails.

Though CH selection phase by new SNs is not associated with the key-establishment phase but is more associated with the cluster-based sensor network architecture. But to support mobility, when an SN or a CH separates from a cluster, the process of CH selection, network registration, and key establishment phases will be iterative. Therefore, we have included the process of CH selection in key establishment phases of the cluster-based sensor network.

### 3.2 Abbreviations and annotations

Table 1 describes the abbreviations and annotations which are used for cryptography and which will be used in the next sections of this paper.

## 4 Secure key establishment method (SKEM)

The proposed key establishment technique is founded on symmetric-key cryptography, which validates the admission of a new node, either an NSN or ASN, and also supports mobility. The proposed algorithmSecure Key Establishment Method (SKEM) will be executed in four phases: The first phase is the Admission of New NSN or ASN and Selection of the parent; the second phase is the Registration Request of New node (NSN or ASN) to Base station via its Parent Node; the third phase is the Authentication of Requesting Nodes (NSN or ASN) by Base-Station, and the last phase is the Concluding process of Key Establishment. The detailed steps of all phases are explained as follows and also in Fig. 2.

### 4.1 Phase-1: Admission of new node (NSN or ASN) and selection of the parent

After positioning the SN in the 2D sensor region, the SN will send a packet known as called “Cluster Head Discovery Request (CHDR)," which contains information like status, degree, and residual energy. The SN will broadcast the CHDR packet at many time instances through a fixed time period,

**Phase-1: Admission of New node (NSN or ASN) and selection of the parent**

**New SN** **Neighbouring NSNs or ASNs**

$CHDR(status,Degree,RE)\ to\ NSN\ and\ ASN \longrightarrow$

$\longleftarrow NHBS\ to\ new\ SN$

***//Parent node chosen based on the algorithm***

**Phase-2: Registration Request of New node (NSN or ASN) to Base station via its Parent Node**

**NSN** **ASN** **BS**

$Mem_{req}=E_{Key_{NSN_i}}(Key_{TS},\ Time-stamp||\ Parent-ID) \longrightarrow$

$Mem_{req}=E_{Key_{new}}\ (Key_{TS},\ Time-stamp||\ 0) \longrightarrow$

**Phase-3: Authentication by Base-Station of Requesting Nodes (NSN or ASN)**

**BS**

$D_{Key_{new}}\ (Mem_{req}) = (Key_{TS}, Time - stamp||\ Parent - ID)$ // ***Authenticate ASN***

$D_{Key_{NSN_i}}(Mem_{req}) = (Key_{NTS}, Time - stamp||\ Parent - ID)$ //***Authenticate NSN***

**Phase-4: Concluding process of Key Establishment**

**For NSN**

**If (NSN is directly linked to ASN)** **// CH will be the parent**

- BS create $E_{Key_{new}}$ with that ASN (Parent)
- $E_{Key_{new}}$ will not be stored in ASN
- ASN will use $Key^{pub}_{ASN-BS}$ with BS to encrypt any sensed data
- BS create $E_{Key_{NSN_i}}$ with NSN's id

**Elseif (NSN is indirectly linked to ASN)** **// Parent is another NSN**

- Parent NSN create $E_{Key_{new}}$ with new NSN
- $E_{Key_{new}}$ will be stored in parent NSN
- New NSN will create $Key_{NSN-BS}$ with BS to encrypt any sensed data
- BS create $E_{Key_{NSN_i}}$ with NSN's id that has chosen as a parent

**For ASN**

BS $\xrightarrow{Acknowledgement}$ new ASNs and its parent.

**Fig. 2** Phases of Proposed SKEM technique

and the duration of this period is dependent on the network parameters and situation. For instance, if the SN's is in a high-density sensing region, we will decrease the length of the sensing period as in high-density sensing regions, the number of neighboring SNs which obtain the CHDR packet is high as compared to low-density sensing regions. Thus, in the high-density region, the new SNs can discover a parent sooner as compared to the low-density region. All the Neighbouring nodes, either NSNs or ASNs, have to respond to the CHDR packet request in the format of the number of hops available

**Table 1** Different abbreviations and annotations used in this work

| Symbol | Meaning/Usage |
|---|---|
| RSA | Rivest–Shamir–Adleman cryptographic algorithm |
| DES | Data Encryption Standard |
| ECC | Elliptic Curve Cryptography |
| AES | Advanced Encryption Standard |
| SHA-1 | Secure Hash Algorithm-1 |
| RAKE | Resource-efficient Authentic Key Establishment |
| CAKE | Codeword Authenticated Key Exchange |
| ECDSA | Elliptic Curve Digital Signature cryptographic |
| ASCW | Algorithm for Wireless-Secure-Communication |
| EDAK | Efficient Dynamic Authentication and Key-management |
| KDC | Key Distribution centre |
| CHDR | Cluster Head Discovery Request |
| SKEM | Secure Key Establishment Method |
| SNS | Normal Sensor Node |
| ASN | Advanced Sensor Node |
| NHBS | Number of hops between the sensor node and BS |
| $E(K, S_d)$ | Encryption of the sensor's data $S_d$ using key $K$ |
| $D(K, S_d)$ | Decryption of sensor's data $S_d$ using key $K$ |
| $F(K)$ | Hash-operation (SHA-1) F, which applied to key $K$ |
| $\|\|$ | Concatenation symbol |

between the sensor node and BS is known as NHBS. The new SN will store the received reply packets obtained from both CHs and SNs and will assign priority to them. Based on the following precedent, the parent node is chosen for a new node:

1. If the new SN is an NSN, then the CH is given the highest priority to be selected as a parent as if an NSN is directly linked to a CH, the energy depletion of other NSNs will not fade away, which are intermediary between them.
2. For the admission of new ASN, only the CHs can be chosen as the parent node as the CH cannot be linked to other CH via NSN. NSN can be connected to one CH only in each communication round.
3. If the new NSN on admission obtains reply messages from more than one CH, then the parent node is selected based on the following principle:

(a) The node with the lowest NHBS vector should, in this case, be chosen as the parent node.
(b) The node with the highest RSSI (received signal strength index) value will be chosen as the parent node when the NHBS format of more than two CHs are the same.

## 4.2 Phase-2: Registration request of new node (NSN or ASN) to base station via its parent node

As discussed in previous Sect. 3.1 also the BS will be the skilled authority to authenticate the requesting SN and CH nodes, and due to specific features of BS in cluster-based sensor network, using it as the capable authority will not introduce any burden. This will not be a performance bottleneck as BS has no energy constraints and is the trusted entity, whereas the CHs are constraint energy resources. The BS is well-thought-out as a Central Skilled Authority for key generation and key-authentication during the admission of new SNs when network overheads exist to manage any overheads. The overwhelmed BS for a scaled sensor network handles and processes network traffic. Also, we have considered several approaches for key establishment methods for NSNs and ASNs because of specific features mentioned about cluster-based sensor networks.

ASN is armed with Tamperproof hardware, which makes them capable of using the key-establishment process with minimum memory requirement. In our proposed method, we have deployed a similar method to wide-ranging network key mechanisms for the purpose of key establishment between the CHs and BS. However, in cryptographic operations, we do not use public shared keys explicitly between CHs and BS to avoid a birthday attack. We created a new key from the mutually shared key between the ASNs and the BS by using a hash-operator $(SHA-1)$ and communicating the ASN's ID of the node. For instance, the key needed for encryption between two CHs with $IDs$ of $CH_1$ and $CH_2$ can be generated by the following hash function: $F(Key^{pub}_{ASN-BS}||CH_1||CH_2)$. Where $Key^{pub}_{ASN-BS}$ is the public key shared between CHs and the BS. Since $Key^{pub}_{ASN-BS}$ is used indirectly by hash functions. It is a one-way function, that is, a function that is practically infeasible to invert; thus, we will be confident that the attackers cannot access $Key^{pub}_{ASN-BS}$. As the exclusive ID of BS in most of the sensor network is considered as 0, So when an ASN wants to submit the registration request to BS, a new key should be created as below:

$$Key_{new} = F(Key^{pub}_{ASN-BS}||CH_{ID}||0) \quad (1)$$

This new key is used for encrypting the request for membership, and then this membership request is then forwarded to the BS. The Membership Invitation message directed to the BS from an ASN should contain the following variable:

1.

Timestamp to avoid the re-play outbreak.
2. The unique CH's ID is nominated as the parent in the preceding phase. If the new node is found to be an ASN and is thus directly connected to the BS (the BS is its parent node), then 0 must be directed as the id of the BS.

$$Key_{TS} = F(Key_{ASN-BS}^{pub}||CH_{ID}||0||Time-stamp) \quad (2)$$

$$Mem_{req} = E_{Key_{new}}(Key_{TS}, Parent-ID, Time-stamp) \quad (3)$$

Though the wide-ranging network key mechanism provides high network flexibility and has reduced memory requirements, we cannot deploy this mechanism for the NSN as the situation and environment are different for them, like they are not armed with Tamperproof hardware. With respect to hardware parameters, if the NSN node is infected and then the keys of other NSNs remain secure, only the key of that node is exposed to the attackers. Thus, we can use rather a different method to the above stated about CHs to create the pre-deployment key, which will be stored in each NSN's memory and is shared with the BS. This implies that we use a key under the supervision of BS to produce the pre-deployment key for NSN; this means that the created keys in higher-level nodes should be computable and stored in lower-level node memory. In other words, when they need them, the nodes at higher levels can produce the necessary keys and do not need to store them to create encrypted communication with the nodes at lower levels. Therefore, memory consumption can be reduced by this function of the key generation. As stated above, we can use a key like $Key_{NSN-BS}$, which is under the supervision of BS for the generation of the shared keys between BS and NSNs to permit the NSNs to encrypt their request for membership and direct them to BS. In this step, we use the hash function's one-way feature and create an NSN pre-deployment key with the NSN ID and store it in its memory:

$$Pre-deploymentkeyforNSN\left(Key_{NSN_i}\right) = F(Key_{NSN-BS}||NSN_{ID}) \quad (4)$$

And even if the BS and the NSN are corrupted, it would always be safe to exchange keys with the other NSNs and the BS. It can therefore be said that a message containing the following two sections should first be created by NSNs to send registration requests to the BS.

1. Timestamp to avoid the re-play outbreak.
2. The Node ID is nominated as the Parent node in the preceding phase.
3. The data is then encrypted with $Key_{TS}$, which is created as $Key_{NSN-BS}$ on pre-deployment.
4. $Key_{NSN_i}$ is generated by the BS and is stored in NSN.
5. $NSN$ compute $Key_{NTS}$ and send $Mem_{req}$ to a base station.

$$Key_{NTS} = F\left(Key_{NSN_i}||NSN_{ID}||Time-stamp\right) \quad (5)$$

$$Mem_{req} = E_{Key_{NSN_i}}(Key_{NTS}, Parent-ID, Time-stamp) \quad (6)$$

### 4.3 Phase-3: Authentication by base-station of requesting nodes (NSN or ASN)

BS is the central skilled authority that has exchanged keys with all SNs (NSNs as well as ASNs) in the proposed process, as mentioned earlier, and is responsible for authenticating the requesting nodes. For secure communication, the BS also gives them a key for encryption and decryption of data shared with their respective parents. As stated in phase 2, several methods for the establishment of the initial shared key with BS are applied to NSNs and ASNs. BS should therefore verify, upon receipt of a demand message from any node, whether that node is NSN or ASN. Therefore, if the message is sent by NSN, BS can retain the shared key obtained from NSN by decrypting the message:

$$\left(Key_{NSN_i}\right) = F(Key_{NSN-BS}||NSN_{ID}) = SharedkeybetweenBSandNSNs \quad (7)$$

Then $Key_{NSN_i}$ is used to decrypt the NSN request for membership:

$$D_{Key_{NSN_i}}\left(Mem_{req}\right) = (Key_{NTS}, Time-stamp||Parent-ID) \quad (8)$$

BS decrypt the NSN request membership with the help of shared key $Key_{NSN_i}$

After the NSN's registration request message has been decrypted, the BS will execute the following operation:

1. *NSN authentication* Because the requesting node, with the exception of BS, will be only SN with $Key_{TS}$. Therefore it will validate the authentication phase.
2. *Registering the requesting NSN parent* This unique ID will be used to create a new key for the NSNs.

When the registration request is received by an ASN, the BS can first create a mutual key between the ASN and itself:

$$Key_{new} = F(Key^{pub}_{ASN-BS}||CH_{ID}||0) \quad (9)$$

BS can now decode the message received by ASN about the membership request. If the received message is decrypted by $Key_{new}$, ASN is authenticated.

$$D_{Key_{new}}(Mem_{req}) = (Key_{TS}, Time-stamp||Parent-ID) \quad (10)$$

### 4.4 Phase-4: concluding process of key establishment

A new key must be created after authenticating the NSN, or ASN has to be exchanged between them and their respective parents. A similar wide-range network key mechanism between ASNs and BS is used as described in phase-2. Therefore, for ASN, no new key should be produced, and this phase is completed by just transmitting a message of receiving to new ASNs and their respective parent. Though, a separate measure can be taken if the new node is an NSN. As mentioned earlier, the characteristics discussed in the literature should have provided keys, which means that they will be generated by higher-level, i.e., SNs on the second tier of the network framework, which is parent nodes and stored in lower-level request nodes, i.e., on the first tier. Memory usage resulting from saving keys can be reduced by this functionality. There are two possibilities if the new node is an NSN:

1. Firstly, if the requesting NSN is directly linked to the ASN and without any intermediate NSN, in this case, the parent will be a CH.
   a BS can first create its mutual key with that ASN in this condition and uses this key to create the new secret key.
   b So, the newly created secret key will not be stored in ASN as whenever the ASN will need it to encrypt any sensed data, it will create its mutual key by using $Key^{pub}_{ASN-BS}$ with the BS; this key will also be used to ASN to create the mutual key with its SN child.
   c BS can then create a new NSN request key with the NSN's id, which has chosen a CH with the ASN's id as the parent node.
2. Secondly, if the requesting NSN is indirectly and with an intermediary NSN linked with a CH, this implies that the parent is another NSN.
   a. The parent NSN will perform authentication and will first create its mutual shared key with that NSN and then uses that key to create the new key.
   b So, the newly created secret key will be stored in the parent NSN; when the new NSN requires the key to encrypt any sensed data, it will create its new shared key with the BS by using $Key_{NSN-BS}$ and then uses $Key_{NSN-BS}$ to create the shared key with its NSN child.
   c BS can then create a new key for requesting a new SN with an NSN's id that has chosen a parent with an NSN's id as the parent node.

## 5 Security investigation of SKEM

In this section, we demonstrate that the proposed SKEM technique is highly secure and is fault-tolerant again major security threats and attacks. The SKEM technique is based on claims that the BS is a trustworthy entity and that the ASNs are armed with hardware that is tamperproof.

Let us assume that the attackers try to introduce their own ASN during the phase of the registration request; they must send an encrypted demand using $Key^{pub}_{ASNs-BS}$. In the request authentication phase of SKEM, the shared secret key is obtained between an ASN's ID, and the BS is attained consequently by $Key_{TS} = F(Key^{pub}_{ASN-BS}||ASN_{ID}||0)$. However, since only BS and other authenticated CH of the network have $Key^{pub}_{ASN-BS}$ and ASN are also armed with Tamperproof hardware, the attackers will not be able to access the $Key^{pub}_{ASN-BS}$; as a consequence, the intruder cannot stop their CHs in the sensor network. The intruders may also aim to attain the keys used in cryptanalytic procedures between CH and BS through a birthday attack in order to steal the $Key^{pub}_{ASN-BS}$. Though, since the $Key^{pub}_{ASN-BS}$ is not used directly in the cryptanalytic procedure. Correspondingly, due to the advantage of the one-way feature of hash function $F(K)$, it is impossible to acquire the $Key^{pub}_{ASN-BS}$.

In case an attacker tries to enter the sensor network as an NSN, then it must encrypt and direct the request message for membership to the BS. The mutual keys which are shared between BS and NSNs are computer as follows: $F(Key_{NSN-BS}||NSN_{ID}) = Sharedkeybetween BSandNSNs$, but since only the BS has access to $Key_{NSN-BS}$, there will be no chances that the attacker can create this shared key. Even in case if the attacker anyhow gets access to the NSN, then also it will only get the data stored in its memory, such as its shared key with the BS, but will not be able to gain access to the $Key_{NSN-BS}$ because of the advantage of the hash functions. Thus, there are no chances to acquire the $Key_{NSN-BS}$ in case of having the shared key of an SN node with the BS. Furthermore, the intruders would not be able to get the other parent node keys by having access to an NSN and acquiring their shared keys with their parent. For instance, if $NSN_i$ is the parent node of $NSN_j$, and the attacker has gained access

to $NSN_j$ and the shared key between $NSN_i$ and $NSN_j$, $F(Key_{NSN[i]-BS}||NSN_j)$; the intruder will not be able to generate the $Key_{NSN[i]-BS}$, that is the mutually shared key in between $NSN_i$ and the BS.

The level of key establishment and authentication is complete between all SNs that will encourage safe and secure data transmission in the proposed work. It should be acknowledged that, since communications are hierarchical in these networks, there is no need for global connectivity in a cluster-based sensor network.

1. To submit registration requests, each SN should have a mutually shared key with the BS.
2. Each SN needs a mutually shared key for data transmission with its parent as well as its child nodes, which are directly linked to it.

If the above preconditions satisfy, then all data transmission will be protected in the sensor network; the proposed SKEM technique perceives these conditions as all network nodes, either NSN or ASN, have a shared key with BS, and it is pre-stored with them before the network initialization. Both NSN or ASN can use this key to create their own mutually shared keys for sharing data with their directly linked child nodes. In addition, after the NSN or ASN are authenticated by BS, a mutually shared key with their parent nodes is created and directed to them. Table 2: depicts the behavior of the proposed SKEM against mischievous attacks. There are many types of attacks that can be performed by an intruder node from which the foremost attacks are: Replay-attack, message manipulation, spoofing, brute force, sinkhole, and selective forwarding, node injection, Sybil attack, and Masquerade attack.

## 6 Performance evaluation

In the previous section, we examined the security of the proposed SKEM technique and explained how it prevents various attacks with a favorable level of security. This section presents the experimental environment and the evaluation results for the proficiency analysis of our proposed work.

### 6.1 Experimental environment

The simulation is performed using the NS2 network simulator, which is presented by Issariyakul et al. [25] to estimate the performance and to authenticate the efficiency of the proposed SKEM technique. BS is located in the center, and NSN is distributed randomly but uniformly in the sensor region. The ASN is arranged in a grid-like structure. We have compared our proposed SKEM technique with EDAK [19], RAKE [13], and RE [12] method. Table 3 specifies the initial value for the simulation parameters.

We have selected these methods due to the presented explanation: The EDAK scheme is proposed for Hierarchical Wireless Sensor Network, and like our technique, it facilitates both the operations of key-establishment and authentication while enhancing the security level. The key distribution algorithm is based on pre-existing information to generate dynamic matrix keys (DMK) and does not need a secure channel and sharing process. In EDAK, the AES algorithm is used for encryption and decryption of the packets between SNs. HSN nodes may execute the MD5 hash function in some specific conditions, i.e., to adapt the pair-wise key dimension of 128bits. The RAKE method is also presented for Hierarchical Wireless Sensor Network with features, and it uses is based on private key cryptography and enables SNs to establish the mutually shared keys for encryption and decryption in an authentic way. Furthermore, RAKE also uses the same cryptographic algorithm ($AES128$bits) and a hash function ($160-$bit$SHA-1$) used in the SKEM technique. The RE method is an ECC-based public-key scheme and used a similar network model as SKEM. RE uses the high memory capacity of HS nodes before sensor deployment to store the public keys of all LS nodes and allows each LS node to retain only the information on its own key.

### 6.2 Results and discussion

The EDAK, RAKE, and RE schemes are compared with the proposed SKEM technique in terms of the issues of energy computation, communication, memory consumption in terms of key storage, scalability, and Complexity. While there is no specific mention of energy here, computation and communication represent its use.

#### 6.2.1 Comparison of communication cost

Initially, it is observed that the energy consumption due to communications in SKEM technique is comparatively higher than EDAK and RAKE as the key generation step in EDAK is based on local prevailing data (DMK) and does not require any key or information which notice reduces the data transmission overhead and improves the security. In the RAKE scheme, the LS- nodes, which are Normal SNs, send a request for membership to the HS-nodes, which are the Advanced SNs, but in the proposed SKEM technique, membership request is sent to the BS. Consequently, more data transmission and communication are needed in SKEM during the key establishment phase. As communications overheads are higher in SKEM, the following advantages are well-known: The ASN has rich resources as compared to NSN, and by increasing energy usage due to communications between ASN, SKEM reduces the storage consumption of NSN in

**Table 2** Discuss how certain possible security threats can be avoided by the SKEM technique:

| S. No. | Name of attack | Description | Prevention from attacks |
|---|---|---|---|
| 1 | Re-play attack | Intruders attempt to retransmit a packet sent by one of the SN's previously, such as a request for membership to the BS, and exhaust the sensor network bandwidth | We have added a timestamp in the Registration Request of New node (NSN or ASN) to the BS via its Parent Node to avoid the re-play outbreak |
| 2 | Message manipulation | Intruders can strive to establish certain service quality disorders by altering the directed messages | Since the SKEM technique supports authenticity, this attack cannot be carried out by the attacker because the key which was used in the cryptographic procedure is required to alter the exchanged messages without finding it out by the receiving nodes |
| 3 | Spoofing attack | To capture essential keys that exchange information, the intruder SN will snoop or listen to the sensor network traffic flow and then acquire a pair-wise secret key | In the SKEM technique, there is no requirement of exchanging any data to create private pair-wise keys. Since both the keys $Key_{NSN-BS}$ and $Key_{ASN-BS}^{pub}$ are used indirectly by hash functions, which is a one-way function, which is practically infeasible to invert, thus we are confident that attackers cannot acquire the keys, which makes it robust against spoofing attacks |
| 4 | Sinkhole and selective forwarding | Intruders try to introduce secretly or inject their own SNs into sensor network architecture by using the node-injection attack | Intruders are not capable of injecting an ASN into the network or of injecting an NSN into the sensor network, as defined in Sect. 5, as it is not necessary for them to establish a common key between them and BS in order to pass the authentication phase and to submit the request for registration to the sensor network |
| 5 | Brute force attacks | This attack is used for the forging of node keys by the attacker node. By checking several possible keys, the attacker attempts to guess the specific SN's secret key | In the SKEM technique, it is impossible for the malicious node to acquire the node keys as SKEM used them indirectly by hash functions, which is a one-way function, which is practically infeasible to invert. Thus, the attackers can't get access to the keys |
| 6 | Masquerade attack | The intruders attempt to impersonate their own sensor nodes instead of authenticated sensor nodes in this attack | This attack is not at all possible for ASN since they are armed with tamperproof hardware, and they do not permit the attacker to access the $Key_{ASN-BS}^{pub}$ that is necessary for an ASN to get authorization in the authentication phase<br>Although the intruders can access one of them in the NSN scenario, they can access their mutual key with BS and can authenticate the malicious node. But in order to mask any NSN of their own, they should have access to one of the authenticated NSNs |
| 7 | Sybil attack | The intruders inject a compromised sensor node with multiple IDs into the sensor network in this attack | This attack on ASN cannot be carried out because, based on previously stated reasons, they cannot be manipulated. However, with regard to NSN, if the intruders are able to access the NSN, they can take admission with their ID and because the mutually shared key of all NSN with BS is created by the NSN's ID. Therefore, if an NSN is added with another ID to the network, the BS cannot authenticate it |

**Table 3** Initial values for simulation parameters and their cost

| Parameter | Symbols | Cost |
|---|---|---|
| Initial energy | $E_{init}$ | 500 J |
| Number of Normal SNs | $NSN$ | 100 to 1000 |
| Number of Advanced SNs | $ASN$ | 5 to 50 |
| Algorithms | – | SKEM, EDAK, RAKE, RE |
| Network field length | $X$ | 500 m |
| Network field breadth | $Y$ | 500 m |
| Simulation time | $t$ | 300 s |
| Transmission energy | $E_{tx}$ | $150\ nJ/sfor 1-bit, 10m$ |
| Reception energy | $E_{rx}$ | $50\ nJ/sfor 1-bit$ |
| Radio electronics energy | $E_{elec}$ | $5\ nJ/bit$ |
| Data Aggregation energy | $E_{da}$ | $5\ nJ/bit/s$ |
| Multipath fading energy | $\varepsilon_{mp}$ | $1.3\ bits/m^4$ |
| Free space amplifier energy | $\varepsilon_{fs}$ | $10\ pJ/bit/m^2$ |
| Type of mobility | – | Random waypoint model |
| Type of traffic | – | Constant Bit Rate |
| Routing | – | Multipath routing model |

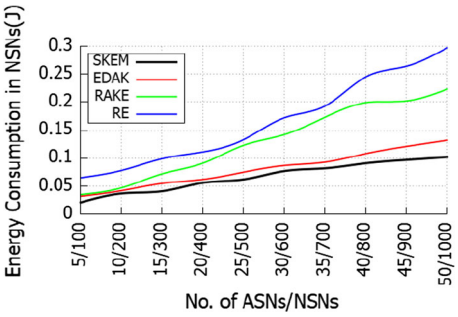


**Fig. 3** Comparison of SKEM with these state-of-the-art techniques for energy consumption resulted from communication in NSNs

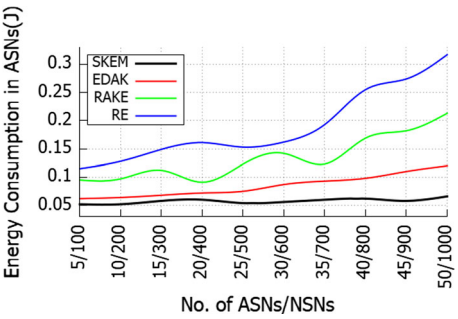


**Fig. 4** Comparison of SKEM with these state-of-the-art techniques for energy consumption resulted from communication in ASNs

a way that only two keys are needed to be stored in the NSN. Storage consumption of ASN is also decreased too.

In EDAK, the admission of new SNs is authenticated and approved at each network restructuring step, and the SN's DMK is re-initialized. This results in more energy consumption.

In RAKE, the parent node selection step for newly entered nodes results in very high energy ingesting in all SN as compared to the complete lifetime of the SKEM technique, as many are located in the route from various SNs to BS. RAKE also has to transmit two extra messages to register and share a new pair-wise key between the SN.

Although the communication with the BS was involved in the key distribution phase in the SKEM technique, the experimental results highlighted that the energy consumption in SKEM under various scenarios is very low as compared to the EDAK, RAKE, and RE schemes for both NSNs as well as ASNs.

A first-order radio communication model developed by Heinzelman et al. [26] is measured used to moderate energy restraint in this work. Figure 3 shows the comparison of the energy consumption of SKEM with the state-of-the-art technique resulted from communication in NSNs. Figure 4 illustrates the comparison of SKEM with these state-of-the-art techniques for energy consumption resulted from communication in ASNs.

### 6.2.2 Comparison of keys storage

The key storage for NSN in SKEM and other compared techniques are presented in Fig. 5. According to this, in SKEM technique have to stores for secure communication, with only two keys for NSN in any network scenarios and with any network size. Figure 6 presents the key storage for ASN in SKEM and other compared techniques. As mentioned in Sect. 4, phase 2 of the SKEM technique, a wide-ranging network key is accomplished for key establishment between the ASN and BS. Therefore, the number of keys stored in ASN for transmission of sensed data to other ASN and BS will always be one, in any network scenarios and with any network size. Also, the ASN does not have to store mutually shared keys with their NSN child as they can create them whenever required. Though the key storage in both LS nodes (NSN), as well as HS-nodes (ASN) in the case of the RAKE scheme, increases abruptly with the network scalability and it is observed as difficult to scale the network as the key storage increases with the increase in the number of SNs in the network.

In the RE scheme, the H-sensors (Advanced SNs) are prestored with the secret key, and all L-sensor (Normal SNs) are also preloaded with one private key. Since L-sensor can identity is equal to the public keys, that means the H-sensor stores $n + 1$ keys, along with some random numbers. After the positioning of the SNs in the RE scheme, the L-sensor

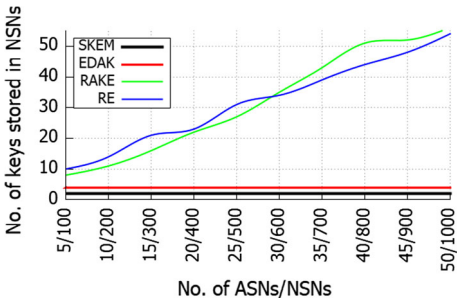


**Fig. 5** Performance comparison of key storage in the NSNs with different NSN and ASN

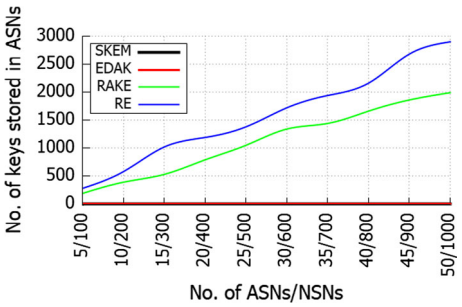


**Fig. 6** Performance comparison of key storage in the ASNs with different NSN and ASN

has to save the private keys shared with other SNs because otherwise, it will be way too expensive to re-generate when required.

EDAK proves to be a storage efficient technique as it needs very less memory area for preliminary keys storage. Certainly, only 4 keys are required for LSNs, and only two keys are required for HSN's nodes, and this independent of any network scenario and network size, which makes it scalable, just like the SKEM technique.

#### 6.2.3 Comparison of algorithmic complexity

SKEM integrates both symmetric-key cryptography and hash function into its architecture, as introduced in Sect. 4 and also examined in Sect. 5, which allows satisfactory usage of characteristics of cluster-based sensor networks to simplify the key establishment phase while accomplishing a higher level of security. For both the phase registration as well as the key establishment, it offers a single protocol and only requires easy encryption and hash calculation. The EDAK protocol is based on the one-way hash calculation and symmetric cryptosystem ($AESandMD5$). It uses two symmetric operations (encryption and decryption) and a maximum of two hash functions. The RAKE and RE methods are both only based on a symmetric key cryptosystem ($AES$) compared to EDAK; RAKE also uses the hash function of $SHA-1$. RAKE protocol, however, incorporates more symmetrical encryption and decryption tree times and performs at least one hash function. Compared with SKEM, EDAK, and RAKE, in the RE technique, along with symmetric key cryptography and hash function, more complex ECC computation is needed. To deal with H–L nodes and H–H nodes, authentications are done as a separate step; it employs two protocols. Therefore, RE is more complex than the other three techniques.

## 7 Conclusion

In this work, we proposed a technique for secure key-establishment based on symmetric cryptography in cluster-based sensor networks; this SKEM technique can reduce energy consumption during communication significantly and provides security against several attacks. By inculcating appropriate solutions in the presented work, we have reduced the storage consumption while securing the network by using only two keys for the NSN and only one key for the ASN. The SNs can communicate horizontally with the SNs at the same level, and also, the SNs can communicate vertically to several levels to achieve complete network coverage without consuming much storage for storing the keys. The proposed SKEM technique is established on the symmetric-key cryptosystem as the energy ingesting due to processing is low, and its storage consumption will be even less than asymmetric key cryptosystem by inculcating suitable procedures. Concerning the proposed solution of the proposed work, the energy ingesting which we have presented in terms of communication cost is reduced outstandingly as compared to related literary works. Furthermore, the SKEM technique also provides higher scalability and resilience against several security attacks.

We intend to expand SKEM in the future to make it accessible to other applications (IoT and biometric) of sensor networks while retaining its merits such as lower storage need, communication efficacy, and scalability; and also, to conduct experiments with real sensor devices.

**Funding** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References


1. Cayirci, E., & Rong, C. (2009). *Security in wireless ad hoc and sensor networks*. Wiley.
2. Bhasin, V., Kumar, S., Saxena, P. C., & Katti, C. P. (2020). Security architectures in wireless sensor network. *International Journal of Information Technology, 12*, 261–272. https://doi.org/10.1007/s41870-018-0103-6
3. Agrawal, M., Zhou, J., Chang, D. (2019) A survey on lightweight authenticated encryption and challenges for securing industrial iot. In: Security and privacy trends in the industrial internet of things. Springer, pp 71–94.
4. Nguyen, K. T., Laurent, M., & Oualha, N. (2015). Survey on secure communication protocols for the internet of things. *Ad Hoc Networks, 32*, 17–31.
5. Mohamed, R. E., Saleh, A. I., Abdelrazzak, M., & Samra, A. S. (2018). Survey on wireless sensor network applications and energy efficient routing protocols. *Wireless Personal Communications, 101*, 1019–1055. https://doi.org/10.1007/s11277-018-5747-9
6. Padmavathi, B., & Kumari, S. R. (2013). A survey on performance analysis of DES, AES and RSA algorithm along with LSB substitution. *International Journal of Science and Research, 2*(4), 170–174.
7. Prakash, S., & Rajput, A. (2018). Hybrid cryptography for secure data communication in wireless sensor networks. In G. Perez, S. Tiwari, M. Trivedi, & K. Mishra (Eds.), *Ambient communications and computer systems. Advances in intelligent systems and computing.* Springer.
8. Singh, S., Sharma, P. K., Moon, S. Y., & Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-017-0494-4
9. Jain, K., Kumar, A., & Vyas, V. (2020). A resilient steady clustering technique for sensor networks. *International Journal of Applied Evolutionary Computation (IJAEC), 11*(4), 1–12.
10. Jain, K., & Kumar, A. (2020). Energy-efficient data-aggregation technique for correlated spatial and temporal data in cluster-based sensor networks. *International Journal of Business Data Communications and Networking (IJBDCN), 16*(2), 53–68. https://doi.org/10.4018/IJBDCN.2020070103
11. Jain, K., & Bhola, A. (2018). Data aggregation design goals for monitoring data in wireless sensor net-works. *Journal of Network Security Computer Net-works, 4*(3), 1–9.
12. Mizanur Rahman, S., & El-Khatib, K. (2010). Private key agreement and secure communication for heterogeneous sensor networks. *Journal of Parallel and Distributed Computing, 70*, 858–870.
13. Shi, Q., Zhang, N., Merabti, M., & Kifayat, K. (2013). Resource-efficient authentic key establishment in heterogeneous wireless sensor networks. *Journal of Parallel and Distributed Computing, 73*(2), 235–249.
14. Khemissa, H., Tandjaoui, D., & Bouzefrane, S. (2017). An ultra-lightweight authentication scheme for heterogeneous wireless sensor networks in the context of Internet of Things. In S. Bouzefrane, S. Banerjee, F. Sailhan, S. Boumerdassi, & E. Renault (Eds.), *Mobile, secure, and programmable networking. MSPN 2017. Lecture Notes in Computer Science.* Springer.
15. Darbandeh, F. G., & Safkhani, M. (2020). A new lightweight user authentication and key agreement scheme for WSN. *Wireless Personal Communications, 114*, 3247–3269. https://doi.org/10.1007/s11277-020-07527-4
16. Jiang, Q., Ma, J., Wei, F., Tian, Y., Shen, J., & Yang, Y. (2016). An untraceable tem-poral-credential-based two-factor authentication scheme using ECC for wireless sensor networks. *Journal of Network and Computer Applications, 76*, 37–48.
17. Li, X., Niu, J., Kumari, S., Wu, F., Sangaiah, A. K., & Choo, K.-K.R. (2018). A three-factor anonymous authentication scheme for wireless sensor networks in inter-net of things environments. *Journal of Network and Computer Applications, 103*, 194–204.
18. Mehra, P. S., Doja, M. N., & Alam, B. (2019). Codeword Authenticated Key Exchange (CAKE) light weight secure routing protocol for WSN. *International Journal of Communication Systems, 32*, e3879. https://doi.org/10.1002/dac.3879
19. Athmani, S., Bilami, A., & Boubiche, D. E. (2019). EDAK: An efficient dynamic authentication and key management mechanism for heterogeneous WSNs. *Future Generation Computer Systems, 92*, 789–799.
20. Mirvaziri, H., & Hosseini, R. (2020). A novel method for key establishment based on symmetric cryptography in hierarchical wireless sensor networks. *Wireless Personal Communications, 112*, 1–19.
21. Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-02020-z
22. Khashan, O., Ahmad, R., & Khafajah, N. (2021). An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks. *Ad Hoc Networks*. https://doi.org/10.1016/j.adhoc.2021.1024
23. Singh, A., Awasthi, A. K., & Singh, K. (2017). Cryptanalysis and improvement in user authentication and key agreement scheme for wireless sensor network. *Wireless Personal Communications, 94*, 1881–1898. https://doi.org/10.1007/s11277-016-3717-7
24. Jain, K., & Kumar, A. (2020). An energy-efficient prediction model for data aggregation in sensor network. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-01833-2
25. Issariyakul, T., & Hossain, E. (2012). *Introduction to Network Simulator 2 (NS2)*. Springer Science+ Business Media. Springer.
26. Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications, 1*(4), 660–670.


**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Akansha Singh** works as an Assistant Professor at Noida Institute of Engineering and Technology, Greater Noida, India. She completed Doctorate from Gautam Buddha University, India. Her research interests include Cryptography, Wireless sensor Networks and Network security.

**Khushboo Jain** works at the School of Computing as an Assistant Professor in DIT University, Dehradun, India. She is pursuing a Ph.D. in Computer Science & Engineering and completed her M. Tech in Software Engineering from Banasthali Vidyapith, Tonk, and B. Tech in Information Technology from MIT, Moradabad. Her research interests include Wireless Sensor Networks, Machine Learning, Data Mining, Data Prediction & Software Engineering. She is an Editorial Board Member and Reviewer of many International Journals. She has published has many research papers in international journals and conferences indexed in SCI and Scopus.