Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5512879, 11 pages
https://doi.org/10.1155/2021/5512879

Research Article

Scalable and Storage Efficient Dynamic Key Management Scheme for Wireless Sensor Network

Vipin Kumar,[1] **Navneet Malik**,[1] **Gaurav Dhiman**,[2] **and Tarun Kumar Lohani**[3]

[1]*Department of Computer Science Engineering, Lovely Professional University, India*
[2]*Department of Computer Science, Government Bikram College of Commerce, India*
[3]*Arba Minch University, Ethiopia*

Correspondence should be addressed to Vipin Kumar; vipin.17730@lpu.co.in

Received 16 February 2021; Revised 4 March 2021; Accepted 21 June 2021; Published 1 July 2021

Academic Editor: Vimal Shanmuganathan

Recently, there have been exploratory growth in the research of wireless sensor network due to wide applications like health monitoring, environment monitoring, and urban traffic management. Sensor network applications have been used in habitat monitoring, border monitoring, health care, and military surveillance. In some applications, the security of these networks is very essential and need robust support. For a network, it is very important that node in the network trust each other and malicious node should be discarded. Cryptography techniques are normally used to secure the networks. Key plays a very important role in network security. Other aspects of security such as integrity, authentication, and confidentiality also depend on keys. In wireless sensor network, it is very difficult to manage the keys as this includes distribution of key, generation of new session key as per requirements, and renewal or revoke the keys in case of attacks. In this paper, we proposed a scalable and storage efficient key management scheme (SSEKMS) for wireless sensor networks that establish the three types of keys for the network: a network key that is shared by all the nodes in the network, a cluster key shared for a cluster, and pairwise key for each pair of nodes. We analysed the resiliency of the scheme (that is the probability of key compromise against the node capture) and compared it with other existing schemes. SSEKMS is a dynamic key management system that also supports the inclusion of the new node and refreshes the keys as per requirements.

1. Introduction

Sensor networks are very popular for collecting information and monitoring activities in hostile areas. In sensor networks, small sensors collect the data like humidity, temperature, pressure, and movements from physical environment. Through a gateway, this information is sent to the sink [1]. A large number of sensors are implemented, and in account of its wireless nature, they easily work in different environmental conditions. Sensors may be deployed in a random manor so it is important to deploy them carefully. If there are a less number of nodes in the area, it may lead to the unattended area or less connectivity of network, and if more nodes are deployed, there will be high traffic in the network and high collision rate of interference between packets. While sending the information to the base station, the security of data is very important and should be implemented properly. To implement the security for this network is a challenging task [2]. Key management is a part of the security technique for a network. The main objective of key management in a sensor network is to maintain the integrity of messages between the communication parties and help to authenticate the nodes in the network. Other than this, the key scheme should be able to deal with node compromise issues and maintain the resiliency of the network against node capture. It should also have a strong node authentication mechanism [3]. These requirements are very important because most of the attacks on wireless sensor networks involve either an inside compromise node or an unauthorized outsider node. In addition, there are other factors as energy and number of messages required for key setup process, scalability, and storage requirement [4]. The rest of the paper is organized as follows: Section 2 explains the security of wireless sensor network and the need of key management and classification

of key management. In Section 3, we have a literature survey of key management schemes for sensor network. Sections 4 and 5 have a network model proposed scheme followed by security analysis of presented work and conclusion at last.

2. Security of Wireless Sensor Network

There is no continuous energy source which powers the sensor nodes. Therefore, proposing energy efficient schemes, which enhance the network lifetime, is another major concern in sensor networks [5]. We proposed a novel storage efficient scheme for enhancing the security and the network lifetime of WSNs. In most of the research on wireless sensor network, security issues are divided into many categories including cryptography, location security, secure routing, secure data aggregation, and secure data fusion [6]. Securing a WSN is a challenging task. Many WSN attacks have been identified by researches which fall under these categories: (i) to manage the keys is an important task to implement and maintain the security of sensor network. Keys are used to encrypt and decrypt the data before sending and receiving. Key can be public and private but it is very important that it must be safe. Public key is known to all but it must be verified that a public key belongs to a legitimate user. Keys are essential to provide authentication, confidentiality, and integrity. Key management is used to allocate and manage keys between network nodes and allows the revocation, updating, and destruction of keys. (ii) Security of routing and routing protocol is also an issue in WSNs. Most of the attacks in network layer make use of authentication loopholes to disturb the processing of message routing [7, 8]. As a result, messages are unable to reach the destination. There are external attackers and internal attacker threats to the routing in WSNs [9]. It is very difficult to identify a compromised internal node because it can generate unauthenticated packets. (iii) To prevent the different attacks on the network like denial-of-service is the next issue. DoS is an organized attack that prevents the user to access the service. It can also increase the delay to access a service. These types of attacks are also very difficult to prevent as data is coming from many sources [10].

2.1. Key Management and Its Need. The distribution of keys is one of the basic problems when security is implemented in WSN. Key management is defined as the set of procedures and techniques to distribute, maintain, and establishment of private key between communication parties [11]. It also includes refresh or update the keys of compromised nodes. It also must maintain forward and backward secrecy. The key management schemes should satisfy the following three groups of metrics [12] which are security, efficiency, and flexibility. If assigned same key for every node and a node is compromised or captured by adversary, it will reveal the key for whole network, and if every node has a different key, then it is very difficult to manage all keys because there are a very big number of node. In case of pool key distribution, if the node has less number of keys, it will create the problem of network connectivity, and to give more number of key to every node, it decreases the resiliency of the network. Public key cryptography usually has the disadvantage of huge resources in demand [13]. Some of the basic requirements of the key are

(i) Public key is very inefficient because it consumes more computation power

(ii) In case of symmetric key approach the main disadvantage is that compromising one node leads to the compromising of entire network

(iii) A predistribution approach requires that pairwise keys are preloaded and then deployed the nodes in the target area. In this approach, every pair has the shared key but this is not found to be suitable for large networks

(iv) There are a number of probabilistic schemes that have some probability of connectivity. In probabilistic scheme, a set of keys are assigned to every node from a large pool of keys and then the probability that two nodes share a common key that can be used as secret key between nodes. To find a compromise between the number of keys assign to each node and total number of keys in key pool is a challenge for large network. A smaller key pool decreases the resilience against node capture attacks while the bigger one reduces the connectivity between nodes because of the reduced probability of share a common key [14]

(v) One challenge in key management is that if a new sensor node joins the network, then, it is difficult in key predistribution case. If it joins the network after key establishment phase, we have to repeat the process

2.2. Types of Key Managements. Many key management techniques have been proposed for WSNs in recent years that can be categorized on different bases like symmetric cryptography or asymmetric cryptography, and other bases are pairwise or GroupWise, centralized or distributed, and dynamic or static [15]. Figure 1 represents the classification of various key management schemes. Various categories of key management are found in literature survey for WSN key management schemes are as follows:

(1) Based on whether to assign pairwise keys or groupwise keys

(a) Pair-wise key schemes vs (b) Group-wise key schemes

(2) Based on whether keys in the nodes are updated or not in network lifetime

(a) Static schemes vs. (b) dynamic schemes

(3) Depend upon the location is known to node or not

(a) Location-dependent vs. (b) location-independent

(4) Based on cryptography techniques

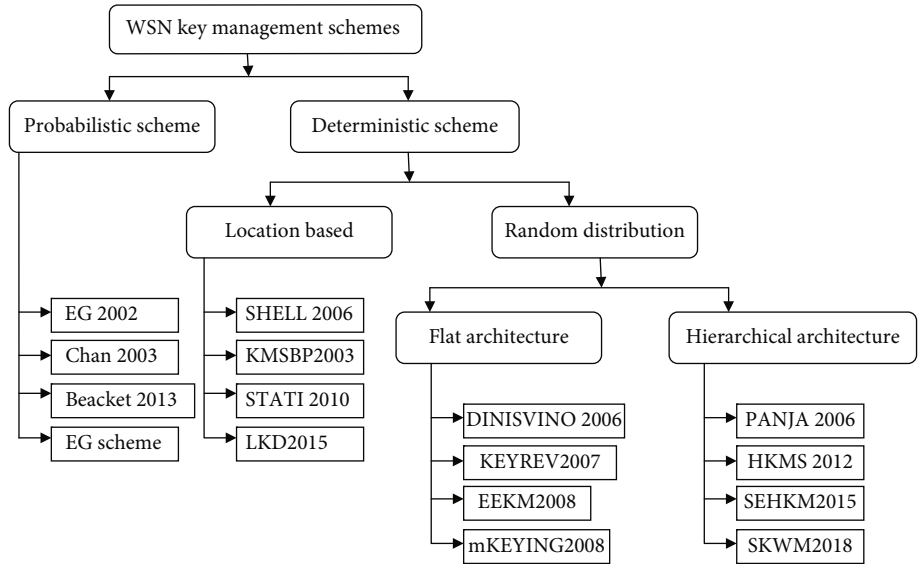

Figure 1: Key management schemes.

(a) Symmetric key scheme vs. (b) asymmetric key scheme

(5) Based on whether the responsibility of key management has been assigned to multiple nodes or a single node

(a) Centralized schemes vs. (b) distributed schemes

(6) Based on memory uses

(a) Storage efficient vs. (b) storage inefficient

2.3. Research Gap. All the schemes for key management proposed in the literature consider the most of the requirements of key management but there is no scheme that considers all the aspects. All the key management schemes previously proposed are lacking in any of area resiliency, single point failure, or scalability. There are three main problems found that must be fulfilled by any key management structure:

Single point failure: scheme must be distributed and the responsibility must be divided to different nodes.

Single key vs. resiliency: if system operates on a single key or a set of keys, it normally does not provide good resiliency against node capture.

Scalability: scalability has been a major challenge for WSN as it consists of a huge number of small sensors. As the number of nodes increases old schemes do not provide support for scalability. In addition to these other challenges include storage efficiency, less energy consumption, and dynamic [16] to support change in network topology.

3. Related Work

Security issues in wireless sensor network achieve a good attention because of its different application [17]. Key management is a fundamental requirement for security of WSN [18, 19]. In this section, we provide work related to manage the keys in sensor network. One of the first key predistribution scheme given by Eschenauer and Gligor known as the E-G scheme [20]. E-G scheme uses the concept of predistribution of keys as a number of keys are preloaded in every node from a common pool of keys. Every node takes a key ring of size m from a big key ring pool of size s. The values of m and s follow the condition $m << s$ means m is very small compare to s. After deployment of sensor node, every node tries to find the common key to every neighbours. If a common key is found between nodes, that key is used as a secure symmetric key. If a common key not found, then, a new key discovery phase started the try to establish the path key using neighbour node. A path key can be established between two nodes if they share the key with third neighbour node. This scheme is very storage efficient and less complex if the values of m and s are chosen carefully. This key management with low processing and storage and complexity requirements is easy to achieve. In this scheme, security is not very good, and connectivity is very poor. It is based on some probability that lies the sharing of common key between nodes.

A new q-composite key scheme which is the extinction of E-G scheme is proposed by Chan et al. [21], says that two nodes can only communicate if they have at least q key common between them, i.e., node must share at-least q keys to communicate. The problem in this scheme is the connectivity as the probability of shared key is reduced. Also, the problem with scheme is if a few nodes compromise the whole network is no thread [22]. Bloom gives a scheme that establishes pairwise keys between nodes. This scheme uses a matrix row and column and by sharing the row and column node can compute common key. The major problem with this scheme is scalability, and matrix is prepared before deployment of node and depends on number of nodes. After deployment, the addition of new node and assignment of keys to new node is not possible. Also, for refreshing the keys, new matrix is needed [23]. These types of key schemes work for homogeneous sensor network that has the same types of node and

does not work for heterogeneous network. If an application requires high level of security; then, these schemes cannot be used as they have various bottleneck in their applications [24]. Therefore, it is of great importance to design an effective key management scheme for homogeneous network as well as heterogeneous sensor network by using the heterogeneity properties [25]. In 2005, Du et al. present a key scheme for pairwise key establishment [26]. This scheme combine the works of the E-G scheme and Blom's work [27] by using the same mechanism as the E-G scheme [20] except the use of individual keys. It uses the k key matrix for each node that is distributed randomly. This process is based on symmetric matrix multiplication, in which $f(i, j)$is deployment on $f(j, i)$. By sending, their identity or partial secret information node can calculate secret common key. Because the scheme preloads the secret information in node in the form of a matrix so this scheme is not supported to at a new node at later stage after deployment. This scheme is not scalable as it is compulsory to create the matrix by number of nodes. Once the matrix is created, it is not possible to add a new entry to the matrix.

A key protocol for the cluster-based network is introduced by Zhu et al.'s LEAP the localized encryption and authentication protocol [28] which is based on a mixed approach and establish the key for every pair of node and a group key for clustered sensor network. This protocol also provides a key that is shared by all the nodes. So for a sensor network, LEAP protocol provides four types of keys that are individual key, pairwise key, group key, and cluster key. Cluster key is the key shared by all the nodes in a cluster, whereas group key is used by sink node to broadcast a secret message to all nodes in the network. Every sensor node can communicate to sink secretly with the help of an individual key that is unique for each node. In this scheme, μTESLA authentication protocol [21] is used to authenticate a node when a message is broadcast by sink. μTESLA is a μTimed Efficient Streaming Loss-tolerant Authentication Protocol used in a broadcast authentication. Leap provides a very good security but it is not storage efficient as it takes a large memory to store the different keys. Also, there is a network key that is deleted after the establishment of a different key, but if at an early stage this key is compromised, every other generated key is known to the adversary.

In 2006, the SHELL protocol is proposed by Tuah et al. [29]. It is location aware and Scalable, Hierarchical, Efficient, and Light-weight (SHELL) protocol that takes the advantage of location awareness of a sensor node. LEAP is a location-based key management protocol, and keys are assigned to node depending on location of each node. This protocol also used multiple types of keys the same as LEAP. There is a key distributor entity for each cluster that has the responsibility for assign the key for each node in the cluster. This scheme has a better resiliency against node capture, but if the key distributor entity captured all the keys in the node revealed, so this protocol has a big drawback as single point failure. The main advantage of the SHELL protocol is high resiliency against node capture. If nodes are deployed randomly, either node has some mechanism to track its location; otherwise, this protocol does not work as it is a location-based protocol.

In 2006, Panja et al. [30] introduced a hierarchical group keying scheme using the Tree-based Group Diffie-Hellman (TGDH) protocol. This is a tree-based protocol, and each key used by the cluster head is made by partial keys of member nodes. This scheme provides an efficient mechanism for rekeying, but as the number of level increases a very high computation required, so this scheme works best for the heterogeneous network with increasing computation power and memory. Adding or removing a node is also a very efficient and simple task as keys are breaking into smaller components.

Das and Sengupta [31] proposed a scheme for a large-scale sensor network key establishment scheme. This scheme is a deterministic scheme and has fixed connectivity and depends on the topology of the hierarchical wireless sensor networks. Although, the storage requirement and computational overhead for this scheme depend on t-degree polynomial, which is a complex process and consume maximum energy. This scheme is very good to establish pairwise key between the neighbours using the same symmetric bivariate polynomials over a finite field.

In 2013, Bechkit et al. propose a new type of Hash-Chain-based key schemes for WSN that uses hash chain to generate new keys [13]. This scheme also needs key predistribution to nodes before deployment. After deployment, new keys are generated using a hash function stored in the nodes. In 2015, Zhang and Wang present a new key management scheme SEHKM that uses the concept of assistant node [32].

Many key schemes for hierarchical sensor network schemes have been proposed in recent years. Zhang and Wang [32] gave SEHKM a secure efficient hierarchical key management scheme. This scheme depends on the Diffie-Hellman key algorithm but it is not scalable as more computation is required for key computation. This scheme is inspired for given less computation and overhead. Messai et al. [33, 34] proposed EAHKM for clustered sensor network, and it is a hierarchical key scheme for clustered sensor network. This scheme is also energy efficient but work only for hierarchical network and does not provide pairwise keys between nodes. The cluster head shared the key with member nodes and not with the other cluster heads. A sequence-based key management scheme is also provided by Messai et al. that uses a sequence number and a not coherent function to generate new keys. It is a series-based key management scheme. In this technique, a big number of partial keys are used which has more storage and communication overhead used by each node.

4. Network Model

The following properties are assumed in regard to the WSN model used in our research, each sensor node has a unique identifier, and all sensor nodes are of the same capability other than base station related to memory, processing unit, and spleen and sensing capabilities. The base station has higher capabilities than others and can communicate using higher range. The sensor node and BS are static after deployment and unaware of their location after deployment and communication is symmetric. If a node si can listen to sj,

Table 1: Notation.

Notation	Description
N_i	ith sensor node in the network denotes the unique id
BS	Base station
CH_i	ith cluster head
k_i	ith key chain in key pool
k_{ij}	j time hash key from ith key chain
h_n	No of time key is hashed
u_0	First term for recursive formula

then, sj is also able to listen si and may use multihope communication. The attacker is assumed to be intelligent and has limited potential. Before taking the full control of network, attacker captures some node and remains invisible. Security scheme must be such that if the behavior of any node is malicious then the key must be renewed from the network. Table 1 represents the notations of various network parameters.

5. Proposed Work

We proposed a decentralized scheme for homogeneous cluster-based architecture. All the key management previously proposed are lacking in any of area resiliency, single-point failure, or scalability. Our scheme is distributed, and there in no single point failure. The resiliency of the scheme is also very good as a set of keys is used by different nodes. Our scheme also supports scalability, and it is storage efficient and also consumes less energy. This is also dynamic and shows the exibility as system topology change. In the proposed scheme, the base station has infinite memory and processing speed, and all sensor nodes are homogeneous and have the same memory and processing speed.

5.1. Key Chain. In our scheme, as shown in Figure 2, we have used the key chain as discussed in [13]. In this method, the base station has a key pool that consists of P noncolliding hash chain of L length and in a single chain every value is considered as potential key. In a chain, the next key is generated by taking the hash of the previous key.

BS randomly selects m chain and assigns m keys to each node before deployment. The key can be hashed at any number of time $l(0¡ = l¡ = L)$. This process is divided into three phases as key predistribution and cluster formation, session key and cluster key generation, and refresh of keys after interval on demand. Two nodes can communicate to each other if they share the keychain.

For renewal of the key or regenerate the new key for new node, a sequence number is also used. Sequence-based key generation is used to generate the keys for those nodes who are not able to find a common key. Each node also has a seed value or first term and a mathematical formula of a sequence. A numerical sequence like $(u_1, u_2, u_3, \cdots)$ is a list of numbers generated for a series. It is a discrete function that, for any integer n, associates a number, denoted un. A recursive formula is used to generate the next term that is used as key.

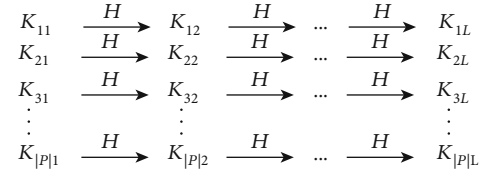

Figure 2: Key chain pool.

Series must be nonarithmetic and nongeometric that is also called nonconvergent series. In the case of nonconvergent series for an attacker, it is very difficult to deduce the values of the sequence terms.

5.2. Key Predistribution and Initialization of Nodes. Whenever setting up a sensor network, various operations need to be performed before and after the network starts its function. The initial network phase has to generate a hash function and hash chain key pool and also generate a nonconvergent recursive formula. Before randomly deployed the sensor network every node is preloaded with three information: a hash chain, first term of sequence, and a recursive function. Hash chain is used to setup a pairwise key to every node, but as key setup using key predistribution, it is a probabilistic scheme and not guarantees every pair shares a common key. If the key is not found between nodes, sequence number and recursive function are used to generate the common key. This phase involves the initialization of node parameters such as node identifiers, transmission range, and keying material. In this phase, all the network nodes are assigned unique identifiers. In addition, each node is assign m keys ($m¡¡P$) from big key pool P and a number $u0$ that tells how many times the key is hashed. A recursive formula is also stored in sensor. Once nodes are initialized, they will be deployed randomly into their target area which they need to monitor.

5.3. Key Setup. After randomly stored, every node sent their id and key chain id to all the neighbour node. If any keys chain matches in both the nodes, they setup the shared keys if no key chain match they use seed value and stored function to generate common key. To generate a common key by using seed value and function, the two nodes send their id and a random number nonce to each other. To maintain the integrity of message, the hash value of the message is also sent. The hash of the message is taken by u_0 times. So a node S_i sends the message $\{S_i \| N_i \| H_{u_0}(S_i \| N_i)\}$, where N_i is the nonce generated by S_i, and u_0 is the first term of nonconvergent formula stored in node S_i. H is the one way hash function. After receiving the message, node S_j generates the secret key by the following process:

(i) Node S_i generates a nonce N_i and send the message $\{S_i \| N_i \| H_{u_0}(S_i \| N_i)\}$ to S_j

(ii) Node S_js generate a nonce N_j and send the message $\{S_j \| N_j \| H_{u_0}(S_j \| N_j)\}$ to S_i

Require: Network
1. Every node S_i broadcast a message Si->∗: M{Si, ki}
2. Upon receiveing the M S_j do following:
    for i and j
    if(ki==kj) list=list+(Si,ki)
    else list=list+(Si,0)
3. Every node S_i store the common key in the list else store 0
4. For every node S_j where Kj ==0 Node S_i send a message
    Si->Sj: M{Si,ui}
5. Upon receiveing M Sj do the following
    if(ki !==0)
    if(uj<ui) H(uj-ui)Ki
else
    generate x= Hu0 (uN1 ‖uN2 ‖S1‖S2)
    list=list+(Ni,x)
6. The node with maximum degree is selected as CH
7. CH generare a group key and distribute to members
8. Every key stored in node is hashed by one more time
Ki=H(ki), u0=u0+1

Algorithm 1: Algorithm for cluster and key setup.

(iii) After receiving the message S_i and S_j computes the uN_i and uN_j and generates the common key $(u_{N_i}\| u_{N_j}\|S_i\|S_j)$

After setup, the pairwise key between every pair of node, a node can send the message to each other using the pairwise key. If the node wants to communicate with the base station, it can use any key of combination of keys and send message and id of keys so that base can use the key to decrypt the message. Once pairwise keys setup all the keys stored, the node is hashed by one more time and this has the same effect as keys are deleted from memory. In this phase node also select cluster head on the bases of node weight that difficult using minimum distance and maximum energy. After election of CH, it generates the group key or cluster key and distributes among the member nodes [29].

5.4. Key Renewal. To increase the life time of network, it is necessary to change the cluster head or change in the topology of network or cluster head is changed key refresh is required. If some node is capture than to isolate the node, we have to change the keys stored in that node. There are different processes to refresh the keys of member node or add a new node in the network. A cluster head refreshes the key any time or as per requirements. For this purpose, CH generates a new group key and distributes among the group nodes encrypted by pairwise key.

5.4.1. If a New Node Is Added to the System. After a long period of time, some node may lose all energy and stop working. New sensor nodes must be added or replace the previous dead node. Our scheme is exible to the addition of new sensor nodes in the network or replace an old node from network and maintain the key of new node. By using the SSEKMS, newly added node is able to share the common key with old node in the network that are previously deployed and neighbours of new node. If a new node is added in the network or replace the old node a set of keys from key pool is loaded to the node and deployed in the field. As base station has the id of the node and list ok key that are loaded in the node it can assign the same key in case of node replacement or give a new set of key in case of new node is added. This node tries to find a common key with neighbour node. If a common key is found between nodes, that key is used as pairwise key, and if no key is found, sequence number and function is used to generate the shared key with neighbours.

5.4.2. Key Refresh for the Node. Whenever any node is compromise or the role of node is change and there is a change in network topology, BS can initiate the key refresh phase. Key can be refreshed for a specific node or for all nodes in the network. Whenever the keys of a specific node is refreshed, a new value of u_0 is sent to node by base station, and this initial term is used to generate the new keys by using Algorithm 1. To refresh the keys for all node in the network base station initiate the key refresh process send a broadcast a message in the network. This message contains the level of node and energy of node and id of the node. Upon receiving of the message every node reset the initial value u_0 according the level of node and execute Algorithm 1 to reset the key. In every round of key refresh, the value of level is start from max value of key chain length.

6. Performance Evaluation

We use the mathematical analysis and simulation process to evaluate the different parameters of our scheme. A theoretical analysis is done to evaluate the parameters, connectivity, storage overhead, and resiliency against node capture. Connectivity of network is given by, that two nodes within the range of each other can send the secure message to each

Require: Network
1. A set of m random keys from key pool S and loaded in the node
2. Recursive function and first term u0 is also stored and deploy
3. First node try to find a common key chain with neighbours by sending it's id Si and key ids stored in this node
4. if node find common key use that as shared key else use random number and function to generate the key
5. Node choose the cluster head as per weightage of nodes

Algorithm 2: Algorithm to maintain the keys of newly added node.

Require: Network
1. BS initiate key refresh by brodcastion message BS->:M{ Hello, BS, Lavel=0, Energy=∞ }
2. All the sensor node receiving the message set u_0=L and forword the message by increasing the level and seting their id and energy lavel.
3. After seting up the value of u_0 all node execute algorithim 1
4. Node chose the cluster head as per weight

Algorithm 3: Refresh the key of node.

Table 2: Comparisons of various schemes.

	Key refresh	Node addition	Location based	Memory efficient
KMP	Yes	No	No	No
DKMM	Yes	Yes	Yes	Yes
EAHKM+	Yes	Yes	No	Yes
SSEKMS	Yes	Yes	No	Yes

other, i.e., they have a shared secret key. Storage overhead and computation overhead are given how much storage and computation power required to perform the operations. Resiliency is also the crucial parameter to check the impact of node capture. A simulation study of scheme is done in NS3 and compares the energy consumed by node or overall network compare to other schemes as shown in Table 2. We implement the scheme NS3 simulator with the following parameters number of nodes 50 to 500 in the area of 300∗ 300 meters with transmission range Tx Range 20DB and a Key pool 1000 key in which 50 keys are assigns to every node.

6.1. Safety Analysis. Sensor networks are also used in far flung areas and deployed in unattended areas to sense various parameters and collect data. So, wireless sensor networks are more prone to various attacks. Two famous attacks are replay attack and node replication. Our scheme is resistant to these attacks, and also resiliency is check against node capture. Resiliency is the probability to reveal the key if node is captured zero resiliency and keys are deleted.

Node replication attack: in node replication, an enemy deploys his own controlled sensor and disturbs the network traffic. For node authentication purpose adversary physically captures a sensor node and extracts all information from sensor. The secret credential stored in node is unveiled to adversary and if adversary captures a large number of nodes many key will be revealed. So key management scheme must be resilient against node capture [35]. To avoid this attack, we deploy different key and key id in every node and base station have id of every allocated key. To apply this attack adversary must have all the keys and can be assign different set of keys to each node.

Replay attack: an active attacker who eavesdrop a valid message between the sender and receiver and send this message at a later time is known as playback attack or replay attack. This can be avoided by time stamp on every message. Every message includes the nonce and a timestamp values and encrypted by the shared key. Receiving node checks the timestamp and nonce value after decryption of message. Receiving node may reject the massage if timestamp is not valid.

6.2. Connectivity. When key setup is completed, a connected graph is created. Two nodes consider as connected if they have a one common key or q common key for q-composite scheme [21]. If every pair in the network has shared key, it is considered as 100 percent connectivity. For probabilistic scheme, connectivity is computed as the probability that every pair in the network shared a key. In case of q composite scheme, each pair must share at least q key so more keys required to increase the connectivity of network. It may increase the resiliency of network but also increase the computation of node and traffic in the network and bandwidth required for message exchange. In the E-G scheme, the probability P_r of sharing at least one key [36] is given by $1 - P$ (node do not share any key) is same as

$$P_r = 1 - \frac{\binom{P}{m}\binom{P-m}{m}}{\binom{P}{2m}}, \quad (1)$$

Where m keys are selected from a pool of P keys, as shown in Figure 3. For q-composite where node share at least

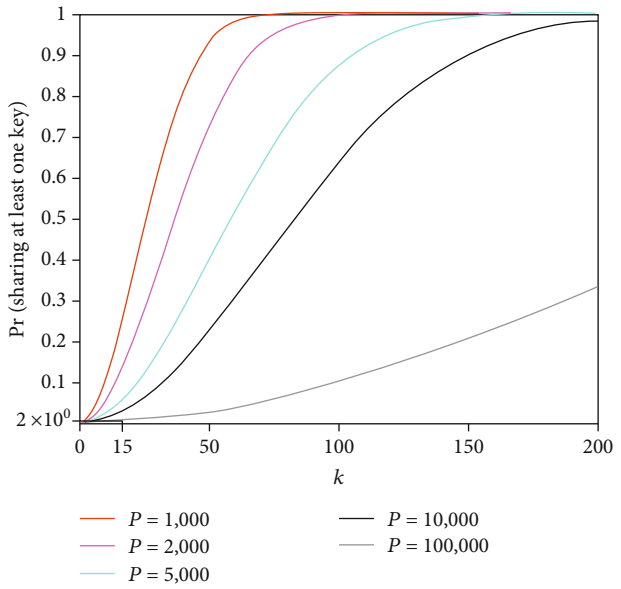

FIGURE 3: Probability of sharing at least one key when two nodes choose k keys from a pool of size [20].

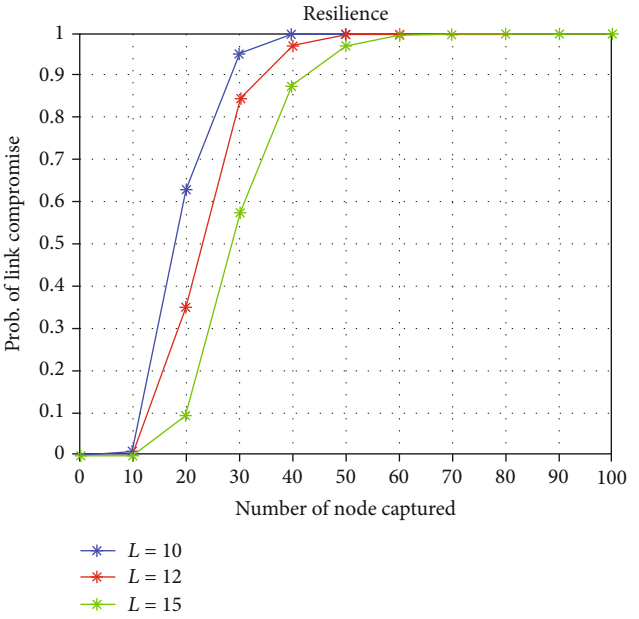

FIGURE 4: Resilience to node compromising attack.

q keys Pr is given by $1 - P$(node share keys less than q). The probability the node share exactly i keys is given by equation

$$P_{\text{SharedExactly}(i)} = \frac{\binom{P}{i}\binom{P-i}{2(m-i)}\binom{2(m-i)}{m-i}}{\binom{P}{m}\binom{P}{m}}, \quad (2)$$

and P_r the probability of shared at least q keys P_r is given by

$$P_r = 1 - (P(0) + P(1) + \cdots \cdots .. + P(q-1)). \quad (3)$$

In our scheme, the probability that every pair of node shared one key is 100 percent as if common key is not found for any pair they can generate the key using seed and function stored in node.

6.3. Communication and Storage Overhead. Sensor node usually has limited memory, around 10 KB [37]. Hence, storing a large number of keys is not desirable. Although storing more number of keys in node increase connectivity of network but it also increase the probability of more key compromise if a node is capture. In over scheme node store only 10 to 20 keys and when common key establish key hashed are stored that have the same effect of key deleted. Other than this, node also stores a variable n and hash function that take negligible space but more secure the network.

6.4. Resiliency of Scheme for Key Compromise. Two important requirements of key management scheme are node authentication and resiliency of scheme. Resiliency is defined as probability of key revealed when a certain part or network is compromised. According to analysis given in [38], when two nodes have the key from same chain, they can share the key with probability $(2i-1)/L2$. For a given key chain, the probability that ith key compromised is $(m/P) * (i/L)$. If this chain is compromised, the probability $P_{\text{ChainComp}}$ is given by:

$$P_{\text{ChainComp}} = \sum_{i=1}^{L}\left(\frac{2i-1}{L^2}\right)\left(1 - \left(1 - \frac{m}{P}\frac{i}{L}\right)^x\right). \quad (4)$$

If a chain is compromised than the fraction of link that uses the chain is given by the ratio of number of links uses that chain to total link establish. So the probability of link compromisation is given by:

$$P_{\text{LinkComp}} = \sum_{i=q}^{m}\left(P_{\text{ChainComp}}\right)^i \frac{P_{\text{Shared}(i)}}{P_{\text{LinkStablish}}}. \quad (5)$$

An important security parameter is the length of chain. The security of scheme is proportional to length of chain but may increase the computation. For better node capture resiliency, the length of chain should be chosen carefully. By increasing the key chain length, more computation is required but it improves resiliency. There is a trade-off between computation and resiliency similar to impact of processing on key chain.

6.5. Expenditure of Energy. We implement the scheme and measure expenditure of energy as shown in Figures 4–6. Due to survive on a battery life sensor nodes, key scheme must not use high energy and should utilize the energy effectively. In this scheme, energy consumption by calculating the average remaining energy in a sensor node and energy required for key establishment. In most schemes, a member node sends a message to head by using several hopes which consume more energy than a single hope

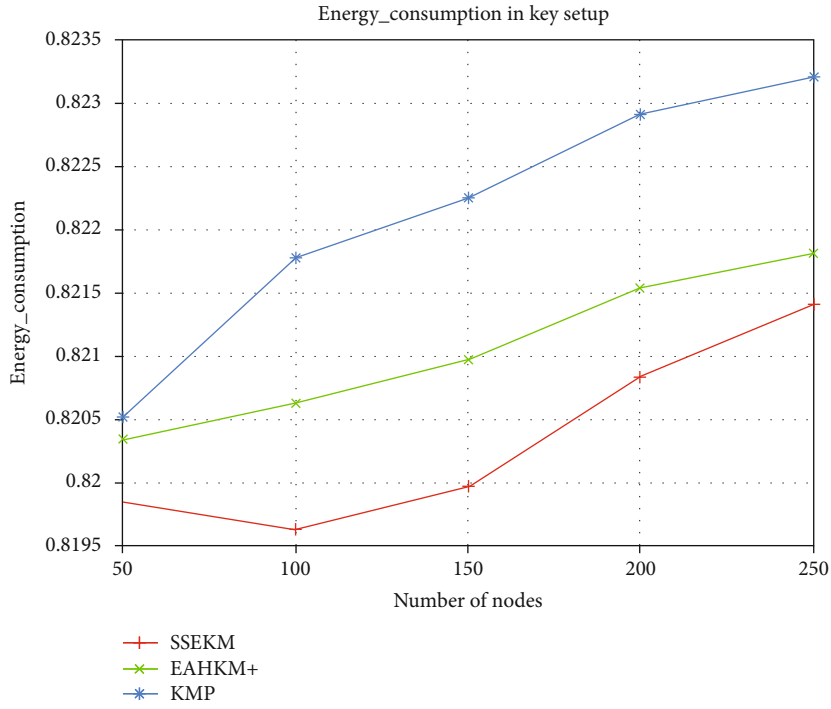

Figure 5: Energy consumption.

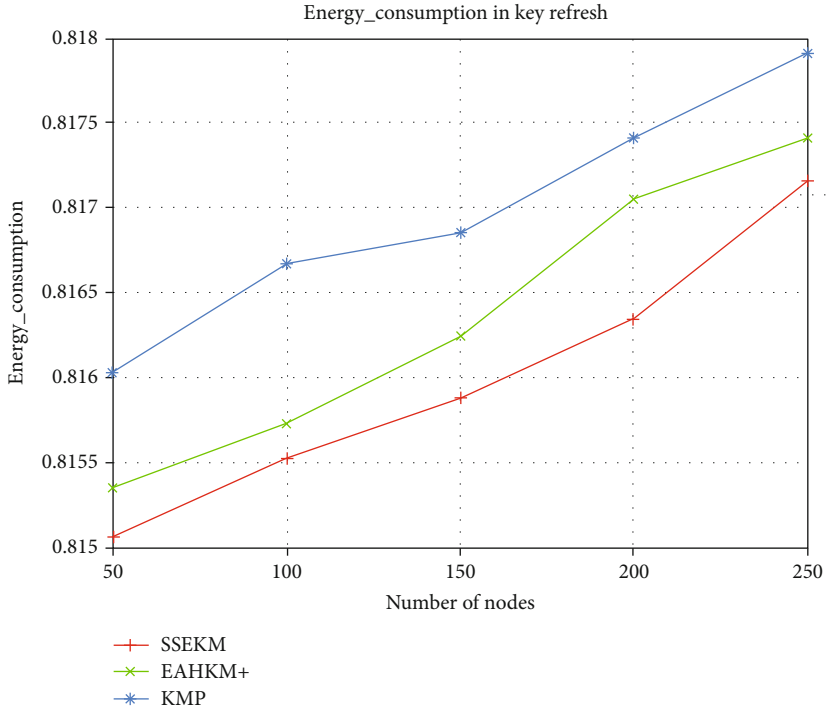

Figure 6: Energy consumption in key refresh.

transmission. In our scheme, some storage is used for storing the key, but to setup the key, only one message per node is required in most of the cases. If key is not setup by recursive formula, then some more energy is required in few of the case.

7. Conclusion

A key management scheme is a vital part of network security. To distribute and manage the key is very important in WSN. The proposed scheme may be extended for other networks

like IoT. In the paper, we presented a storage efficient dynamic key management technique (SSEKMS) for pairwise key distribution. These keys also support for secure cluster formation and secure connection with base station for each node. Additionally, SSEKMS supports key refreshment and key revoking in the network. SSEKMS has sequence-based key generation for refreshing the keys and used key predistribution with hash chain keys. With respect to other old schemes, our scheme is storage efficient and secure against node capture resiliency. It is also energy efficient compared to other schemes like LEAP, SKM, and SKWN. Unlike SHELL which is location-based and needs location information in advance, our scheme does not need any prior information and works for random distribution. This works in both the cases as pairwise key as well as groupwise key. In the future, we want to extend our scheme for heterogeneous systems and IoT security [39]. In the future, this technique can be extended to support modern networks, which could involve heterogeneous IoT networks [16]. Further use of artificial intelligence and optimization can be found in [36, 40–42].

Data Availability

The data is open and shall be made available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[3] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security (final)," *DARPA Project report*, vol. 1, Tech. Rep. 1, Cryptographic Technologies Group, Trusted Information System, NAI Labs, 2000.

[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrish-nan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*, p. 10, Maui, HI, USA, 2000.

[5] X. Zhang, H. M. Heys, and C. Li, "Energy efficiency of symmetric key cryptographic algorithms in wireless sensor networks," in *2010 25th Biennial symposium on communications*, pp. 168–172, Kingston, ON, Canada, 2010.

[6] A. K. Gautam and R. Kumar, "A comparative study of recently proposed key management schemes in wireless sensor network," in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 512–517, Greater Noida, India, 2018.

[7] G. Dhiman and M. Garg, "Mosse: a novel hybrid multi-objective meta-heuristic algorithm for engineering design problems," *Soft Computing*, vol. 24, no. 24, pp. 18379–18398, 2020.

[8] H. Kaur, A. Rai, S. S. Bhatia, and G. Dhiman, "Moepo: a novel multi-objective emperor penguin optimizer for global optimization: special application in ranking of cloud service providers," *Engineering applications of Artificial Intelligence*, vol. 96, article 104008, 2020.

[9] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[10] M. Sharma, A. Tandon, S. Narayan, and B. Bhushan, "Classification and analysis of security attacks in wsns and ieee 802.15.4 standards: a survey," in *2017 3rd International Conference on Advances 21 in Computing, Communication & Automation (ICACCA)(Fall)*, pp. 1–5, Dehradun, India, 2017.

[11] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC press, 2018.

[12] M. A. Simplício Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer networks*, vol. 54, no. 15, pp. 2591–2612, 2010.

[13] W. Bechkit, Y. Challal, and A. Bouabdallah, "A new class of hash-chain based key pre-distribution schemes for wsn," *Computer Communications*, vol. 36, no. 3, pp. 243–255, 2013.

[14] S. A. Camtepe and B. Yener, "Key Distribution Mechanisms for Wire-Less Sensor Networks: A Survey," Rensselaer Polytechnic Institute, Troy, New York, Technical Report, 2005.

[15] J. Zhou, Z. Cao, X. Dong, N. Xiong, and A. V. Vasilakos, "4s: a secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks," *Information Sciences*, vol. 314, pp. 255–276, 2015.

[16] G. Dhiman and V. Kumar, "Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications," *Advances in Engineering Software*, vol. 114, pp. 48–70, 2017.

[17] D. Qin, S. Jia, S. Yang, E. Wang, and Q. Ding, "Research on secure aggregation scheme based on stateful public key cryptology in wireless sensor networks," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 7, pp. 938–948, 2016.

[18] K. Lu, Y. Qian, M. Guizani, and H.-H. Chen, "A framework for a distributed key management scheme in heterogeneous wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 639–647, 2008.

[19] J. Metan and K. N. N. Murthy, "Robust and secure key management in wsn using arbitrary key-deployment," in *2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pp. 246–250, Mandya, India, 2015.

[20] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 41–47, New York, NY, USA, 2002.

[21] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *2003 Symposium on Security and Privacy*, pp. 197–213, Berkeley, CA, USA, 2003.

[22] M. Rahman, S. Sampalli, and S. Hussain, "A robust pair-wise and group key management protocol for wireless sensor network," in *2010 IEEE Globecom workshops*, pp. 1528–1532, Miami, FL, USA, 2010.

[23] C. Yang, J. Zhou, W. Zhang, and J. Wong, "Pairwise key establishment for large-scale sensor networks: from identifier-based

to location-based," in *Proceedings of the 1st international conference on Scalable information systems*, p. 27, New York, NY, USA, 2006.

[24] V. P. Mhatre, C. Rosenberg, D. Kofman, R. Mazumdar, and N. Shroff, "A minimum cost heterogeneous sensor network with a lifetime constraint," *IEEE Transactions on Mobile Computing*, vol. 4, no. 1, pp. 4–15, 2005.

[25] V. Sharma and M. Hussasin, "Node authentication in wsn using key distribution mechanism," in *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, pp. 1–7, Indore, India, 2016.

[26] D. Wenliang, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," in *ACM Transactions on Information and System Security (TISSEC)*, vol. 8no. 2, pp. 228–258, New York, NY, USA, 2005.

[27] R. Blom, "An optimal class of symmetric key generation systems," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 335–338, Springer, 1984.

[28] S. Zhu, S. Setia, and S. Jajodia, "LEAP+," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[29] N. Tuah, M. Ismail, and K. Jumari, "Evaluation of optimal cluster size in heterogenous energy wireless sensor networks," in *2012 International Symposium on Telecommunication Technologies*, pp. 124–130, Kuala Lumpur, Malaysia, 2012.

[30] B. Panja, S. K. Madria, and B. Bhargava, "Energy and communication efficient group key management protocol for hierarchical sensor networks," in *IEEE International Conference on Sensor Networks,Ubiquitous, and Trustworthy Computing (SUTC'06)*, vol. 1, p. 8, Taichung, Taiwan, 2006.

[31] A. K. Das and I. Sengupta, "An effective group-based key establishment scheme for large-scale wireless sensor networks using bivariate polynomials," in *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, pp. 9–16, Bangalore, India, 2008.

[32] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks," in *2015 International Conference on Computing, Communication and Security (ICCCS)*, pp. 1–7, Pointe aux Piments, Mauritius, 2015.

[33] M.-L. Messai and H. Seba, "Eahkm+: energy-aware secure clustering scheme in wireless sensor networks," *International Journal of High Performance Computing and Networking*, vol. 11, no. 2, pp. 145–155, 2018.

[34] M.-L. Messai, H. Seba, and M. Aliouat, "A new hierarchical key management scheme for secure clustering in wireless sensor networks," in *International conference on wired/wireless internet communication*, pp. 411–424, Springer, 2015.

[35] W. T. Zhu, J. Zhou, R. H. Deng, and F. Bao, "Detecting node replication attacks in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 1022–1034, 2012.

[36] G. Dhiman and V. Kumar, "Emperor penguin optimizer: a bio-inspired algorithm for engineering problems," *Knowledge-Based Systems*, vol. 159, pp. 20–50, 2018.

[37] T. Ian, "Downard. Simulating Sensor Networks in ns-2," Technical report, Naval Research Lab, Washington DC, 2004.

[38] V. Maty and P. Venda, "Two improvements of random key predistribution for wireless sensor networks," in *International Conference on Security and Privacy in Communication Systems*, pp. 61–75, Springer, 2012.

[39] S. Vimal, A. Suresh, P. Subbulakshmi, S. Pradeepa, and M. Kaliappan, "Edge computing-based intrusion detection system for smart cities development using iot in urban areas," in *Internet of things in smart Technologies for Sustainable Urban Development*, pp. 219–237, Springer, 2020.

[40] G. Dhiman and A. Kaur, "Stoa: a bio-inspired based optimization algorithm for industrial engineering problems," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 148–174, 2019.

[41] M. Garg and G. Dhiman, "Deep convolution neural network approach for defect inspection of textured surfaces," *Journal of the Institute of Electronics and Computer*, vol. 2, no. 1, pp. 28–38, 2020.

[42] P. Singh and G. Dhiman, "A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches," *Journal of Computational Science*, vol. 27, pp. 370–385, 2018.