Cluster Based Secure Dynamic Keying Technique for Heterogeneous Mobile Wireless Sensor Networks

Thiruppathy Kesavan. V[1*], Radhakrishnan. S[2]

[1] Department of Computer Science & Engineering, Madanapalle Institute of Technology & Science, Madanapalle, Andhra Pradesh, India
[2] Department of Computer Science & Engineering, Kalasalingam University, Krishnankoil, Tamil Nadu, India

Abstract: In Heterogeneous Wireless Sensor Networks, the mobility of the sensor nodes becomes essential in various applications. During node mobility, there are possibilities for the malicious node to become the cluster head or cluster member. This causes the cluster or the whole network to be controlled by the malicious nodes. To offer high level of security, the mobile sensor nodes need to be authenticated. Further, clustering of nodes improves scalability, energy efficient routing and data delivery. In this paper, we propose a cluster based secure dynamic keying technique to authenticate the nodes during mobility. The nodes with high configuration are chosen as cluster heads based on the weight value which is estimated using parameters such as the node degree, average distance, node's average speed, and virtual battery power. The keys are dynamically generated and used for providing security. Even the keys are compromised by the attackers, they are not able to use the previous keys to cheat or disuse the authenticated nodes. In addition, a bidirectional malicious node detection technique is employed which eliminates the malicious node from the network. By simulation, it is proved that the proposed technique provides efficient security with reduced energy consumption during node mobility.

Keywords: cluster; dynamic key; heterogeneous; mobility; wireless sensor network; weight

I. INTRODUCTION

The Heterogeneous Wireless Sensor Network (H-WSN) [1] [2] involves dissimilar sensor nodes having different capabilities like longer transmission capacity, more memory and processing capability. The mobility of the sensor nodes becomes essential in many applications. To improve the scalability of the network, clustering should be performed. Due to the movements of the nodes, the clustering has to be performed dynamically. Traditional clustering algorithms like Hybrid, Energy-Efficient, Distributed Clustering (HEED) [3], Energy efficient heterogeneous clustered scheme (EEHC) [4], Stochastic Distributed Energy-Efficient Clustering (SDEEC) [5], Unequal clustering algorithm [6] are subject to various attacks on clustering. The H-WSNs are utilized for various real time applications including event detection, localization, health care monitoring [7] [8], transportation system [9] and multimedia applications which can function in diverse environments. Providing security in WSNs is a challenging factor due to the resource constraints in sensor nodes as well as

A secure mobility aware dynamic keying technique for authentication in WSN is proposed in this paper.

the size and density of the networks. Since the sensor nodes are deployed in unattended area, providing physical security becomes impossible. Since the topology is mainly based on the network deployment, it cannot be predicted. Data confidentiality, integrity, authenticity and availability are considered to be the security requirements which the WSN should provide even in the presence of powerful attackers [10].

The threats and challenges that occur in H-WSN are disruption of cooperative transmissions which include selfish behaviour, routing attacks, jamming of signals and other attacks such as Sybil attack; Wormhole attack; rushing attack; resource draining attack; relay discovery attack; bad mouthing attack; traffic injection attack and query flooding attacks [11].

Many nodes in the network share a secret encryption key which is termed as session key. Several situations occur where a sensor node is forced to leave the network. One among the situations is due to the complete exhaustion of battery power or due to its malicious behaviour. If a node is forced to leave the network, all the keys that are known to that node have to be changed. At this stage, re-keying has to be securely performed using the other administrative keys and the new keys should not be known to the compromised node [12]. Re-keying has to be performed during topology changes, due to node failure, node addition and node compromise [13] [14]. The pairwise keying model is more robust against the node capture attack because compromising a single node does not affect the other nodes in the network. In this model, every node has to maintain $N - 1$ keys where N is the number of nodes in the network. Due to this, scalability has become an issue because the number of keys to be maintained in every node is directly proportional to the network size. It requires more storage space if the network size increases [10] [15] [16] [17].

Authentication is one of the security requirements that verify the identity of a sensor node. The Public key cryptography is the natural method for providing authentication in a large network of sensor nodes. The drawback in using public key cryptography is that, it consumes more power. As a result, the sensor nodes use Elliptic Curve Digital Signature Algorithm to generate digital signature for authentication [18]. The combination of pairwise key, global key, cluster key and preloaded secret information is also used for authenticating the nodes in the network [13] [19]. In this proposed technique, we have adopted Hashed Message Authentication Code (HMAC) algorithm [20] for authenticating the nodes inside the network.

The node mobility results in random topological changes which occur regularly. Due to this, the security is affected in mobile network environment. The security requirements [21] to be satisfied while designing a technique are:

***Freshness of key*:** Due to the node mobility, the key(s) need to be updated and made available among all the Sensor Nodes (SNs) and the Cluster Head (CH).

Authentication of nodes: Any node can join with or leave from any CH. Similarly, any heterogeneous node can become a CH for a cluster. Due to these dynamic changes, the nodes need to be authenticated.

Preserving data integrity: The data should not be altered by any adversary due to the topology changes.

In this paper, a framework has been derived for key management during SN and CH mobility. During message communication, the authenticity is verified and processed. To provide integrity for the data, the RC5 symmetric cryptographic mechanism [22] is adopted.

1.1 Problem definition

A Cluster based Secure Dynamic Keying Technique (CSDKT) for authentication in WSN has been proposed in [23]. This technique determines the dynamic key for static network where the node location does not change. But, many applications require node mobility support in the network. In such mobile network, there are possibilities for a node to detach from the existing cluster and attach

to another cluster. The detaching node will be either a CH or cluster member. The main cause for the changes in the CHs and cluster memberships is mobility. The mobility of nodes coupled with the transient nature of wireless media often results in a highly dynamic network topology. In such a case, security has to be incorporated in the network. In this work, a mobility aware dynamic keying technique is designed for authenticating the nodes in H-WSNs.

The rest of this paper is organized as follows. In Section 2, the related works are discussed. Section 3 describes the cluster based secure dynamic keying technique which includes dynamic clustering, dynamic key management, authentication technique and bidirectional malicious node detection technique. Section 4 presents the performance evaluation of CSDKT and finally, section 5 concludes the paper.

II. Related Works

Hong [24] has proposed a weighted clustering algorithm for clustering in a mobile WSN. The CHs are chosen based on the weight value that is calculated by the parameters such as degree difference of a node, sum of the distances between a node and its neighbors, mobility speed, characteristic of a sensor node, the number of times a node acted as CH. The node with minimum weight value is chosen as CH. The weight value is calculated for every fixed time interval. In this method, if a CH moves from one location to another immediately after it is chosen as CH, the member nodes that are associated with the CH could not be able to send the data to the BS. For H-WSN, Dilip Kumar et al. [4] and Elbhiri et al., [5] have proposed Energy Efficient Heterogeneous Clustered (EEHC) and Stochastic Distributed Energy-Efficient Clustering (SDEEC) schemes respectively. EEHC is based on weighted election probabilities of each node to become a cluster head according to the residual energy in each node. In SDEEC also, all the nodes use the initial and residual energy level to define the CHs. HEED is another dynamic clustering algorithm proposed by Ossama and Fahmy [3] is an extension of basic LEACH protocol [25] that uses the residual energy and node density for clustering.

The Localized Combinatorial Keying (LOCK) [26] proposed by Mohammed Eltoweissy et al. is a Exclusion-Based Systems (EBS) [27] dynamic key management method for group based sensor networks. The LOCK uses three types of keys which include administrative keys; group session keys and cluster session keys. The Special nodes called as the key generation nodes elected by cluster leader perform the key generation process. The LOCK is meant for static networks. But the proposed technique supports node mobility. LOCK uses polynomial based key generation scheme where the number of unknown (m) polynomial keys are directly proportional to network resilience. But the value of m is inversely proportional to network connectivity. The Higher number of polynomials per node increases the connectivity of the network. It reduces the network resilience if node capture attack happens. LOCK requires re-keying if the probability of node capture reaches a particular threshold level. Thereby re-keying requires m number of messages to be transmitted. Since LOCK has 3 different types of keys to maintain, naturally it has the higher overhead compared to the proposed scheme. When compared to LOCK, the proposed scheme can be applied in mobile nodes and does not involve any additional nodes like key generation nodes apart from the cluster heads. The CHs are involved in key management and coordination process. Ozgur [28] has proposed a multi-level dynamic key management scheme using Unmanned Aerial Vehicle (UAV) which acts as a key distribution and coordination center for asymmetric keys. This scheme constructs symmetric keys using asymmetric keys for its further communication. This scheme highly depends on UAV which is vulnerable to physical attack. In this scheme, if a node is compromised, the public key of the neighboring node can be obtained and it can be used for

illegal operations in the network. But the proposed technique does not depend on a single device for key distribution and coordination. The CHs are involved in key management and coordination process.

Shu and Meng-Hui [29] have proposed two group key management schemes such as Hybrid Group Key Establishment Scheme and Group Key Establishment Scheme with Initial Shared Keys for hierarchical WSN. They have combined both the symmetric technique for data encryption and integrity and the asymmetric technique for signatures and key management operations. For both the schemes, they have analyzed the communication and storage overhead. This kind of architecture is suitable for the applications where power source is always available. Moreover, this scheme is not much scalable because the communication overhead at Forwarding Node is very high. Also this scheme requires an initially preloaded secret key K_s which is not necessary in our work. Qiuhua Wang et al. [30] have proposed a self-healing group key distribution scheme (SGKD) based on one-way hash chain with the capability of recovering previous group session keys. This can be done by the users joining the group in different sessions and they are treated differently by binding the time at which the user joins the group. This scheme is capable of resisting the network from the collusion attack using the unique joining identity for each session. Tim Landstra et al. [31] have proposed a static-dynamic key management protocol that utilizes the distinct security requirements of WSN. This protocol uses Self-organizing protocol to create sub-network around an event. The majority of the network in this protocol operates in low security mode with static keys to save energy. The remaining part of the network or sub-network operates in high security mode with dynamic keys. For creating dynamic sub-network keys, this protocol uses a modified version of the protocol proposed in [32]. They have also proposed a protocol that revokes a node from the network and updates the network and node's individual keys. But this protocol uses both the static and the dynamic key management approaches which increase the storage and communication cost, which are less in the proposed scheme.

Pardeep Kumar et al. [33] have proposed a mutual authentication and dynamic key establishment scheme that suits for real time heterogeneous WSNs. In this scheme, mutual authentication happens during cluster formation and new node addition. The Dynamic session keys are generated using the dynamic secret number which is generated dynamically by H-sensors and the dynamic session key is used for authentication. Qiu et al. [34] have proposed an authentication scheme for dynamic WSN for large scale distributed WSN. When a node moves inside the network, it has to send a request message to the Base Station (BS). The BS validates the message and sends a session key to the mobile node through CH. Qiu scheme guarantees that any two SNs share at least one key with the probability of 1 with minimum energy and memory overhead. In this scheme, since the re-authentication depends on the BS, it increases the communication overhead. Since each node maintains a key cache, the keying overhead will be more. But in our work, the re-authentication does not depend on BS and no key cache is maintained for nodes.

Many authors have proposed malicious attack detection techniques on cluster based WSNs. Shun-Sheng Wang et al. [35] have proposed an Integrated Intrusion Detection System (IIDS) intended for the sink, cluster head and sensor node relying on the different capabilities. In addition, there is an anomaly and a misuse detection module is proposed in IIDS by which the detection rate is enhanced and the false positive rate is reduced. However, the feature selection method affects IDS performance. Xiao Zhenghong and Chen Zhigang [36] have proposed another IDS which utilizes an energy prediction model for detecting attacks in cluster head election phase. However, the energy consumption is doubled in a period of time. In CSDKT, both malicious cluster members as well as cluster heads are identified.

From the above discussion, we can summarize the issues of existing works like increased storage overhead, communication overhead, applicable only for static networks, depending on any centralized authority or device, having static or preloaded keys etc. The proposed solution aims to solve the above mentioned issues. The proposed technique can be applied for mobile scenario since it considers the location of nodes in the dynamic key generation process and it does not incur high overhead. It does not depend on a single device for key distribution and coordination since only the cluster heads and sink are involved in the key generation process.

III. Cluster Based Secure Dynamic Keying Technique

3.1 Overview

In this paper, cluster based secure dynamic keying technique is proposed for authentication in H-WSN that supports mobility and malicious node detection technique. The notations used in this paper are listed in Table I.

In the CSDKT, once the nodes are deployed in the field, the high power heterogeneous nodes enabled with GPS are selected as CHs based on the weight values that are dynamically calculated. The nodes with minimum weight values are chosen as CHs. The weight value is estimated based on the dynamic parameters such as *ND, D_{av}, S_{av}* and V_{BP}. Once the CHs are selected, each node in the network has to calculate a value called Combined Cost Value (CCV). The CCV has to be used by the nodes to derive the dynamic key which is to be used for protecting the data in the upcoming communications. The dynamic keys that are calculated by the nodes are updated regularly based on the changes in the CCV value. When the CH tends to leave the network, the CH re-election is performed by the cluster members. To ensure the security among the network, the mobility of cluster members and CHs are performed by authentication using the dynamic key. A bidirectional malicious node detection technique is employed to eliminate the malicious cluster members and CHs from the network. The flowchart for the overall process is depicted in Fig. 1.

3.1.1 Network Model

We represent the H-WSN which supports node mobility inside the network. The network consists of BS (also said as sink), CH and resource-constrained SNs. Among these, the BS

Table I *Notations*

Notation	Description	Notation	Description
N_i	Node *i*	DRL	Dynamic Reference Localization
K_D	Dynamic Key	$E(K_D,M)$	Message Encryption by key K_D for message M
C_i	Cluster *i*	HMAC	Hashed Message Authentication Code
W_i	Weight of node *i*	REVMES	REVocation MESsage
GPS	Global Positioning System	CHIC	CH Invalidation Command
M_e	Encrypted message	$H(K_D,M)$	HMAC function for message using K_D
D_{av}	Average distance with its neighbours	TT_{BS}	Threshold Trust Value set by BS
S_{av}	Average Speed	HELLO	Hello Message
ND	Node Degree	ML_i	Malicious Node $_i$
NL	Node Location	CH_INFO	CH Information Message
V_{cs}	Virtual Cost at source node	CH_NOM	CH Nomination Message
V_{ci}	Virtual Cost at intermediate node	CL_JOIN	Cluster Join Message
V_{BP}	Virtual Battery Power	NOD_WGT	Node Weight Message
CM_i	Cluster Member *i*	$T^X_{ij}(t)$	Trust value of Component *X* between *i* and *j* at time interval *t*
TT_{CH}	Threshold Trust Value set by CH		

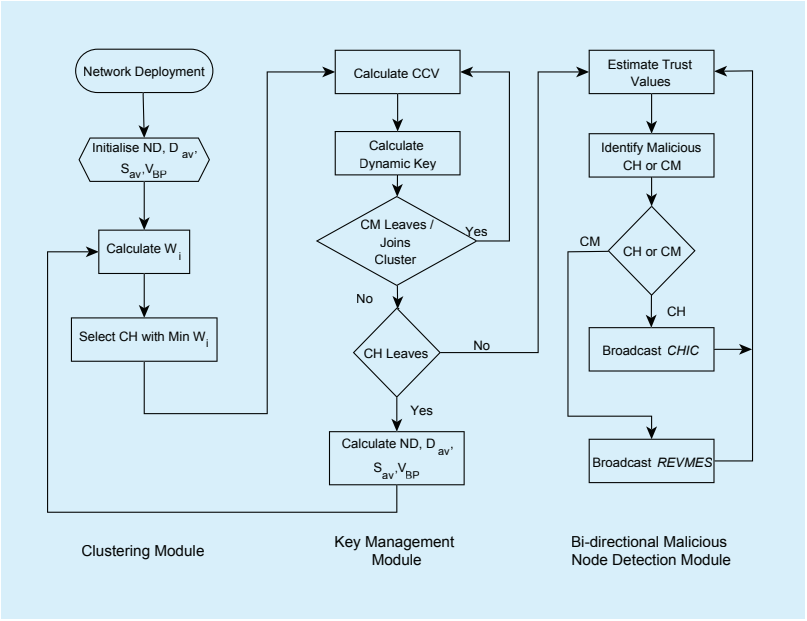

Fig.1 *Flow Chart of the overall process*

Table II *Assumed power consumption for calculating* V_{BP}

Notation	Description
P_{rx}	Reception Power
P_{tx}	Transmission Power
P_{enc}	Encoding Power
P_{dec}	Decoding Power
P_a	Power required to maintain the node in the Alive state
P_{sync}	*Power utilized to Synchronize the source* $(Z \times (P_{rx} + P_{dec}) + t \times P_a)$
P_{MAC}	Power required to generate MAC code
P_{auth}	Power required for Authentication

and CHs are more powerful in terms of energy, memory and processing ability compared to SNs. The numbers of CH varies based on the size of the network. The SNs collect information based on the application and transmit the readings to the CH. When a CH needs to send data to the sink, it splits the packet into various numbers of shares by threshold secret sharing algorithm [14] and forwards them to the sink via multiple paths. This routing technique is called as Multi-path dispersal routing [23]. In this network, the BS is static, CH and SNs can move inside the network. The users of the application communicates to the network from BS where it collects the data send by CHs, performs analysis, if required, sends commands to SNs through CHs and identifies malicious nodes among CHs.

3.2 Dynamic clustering

3.2.1 Estimation of virtual battery power

The V_{BP} is considered instead of the real battery power because the differences in the power range across the nodes which induce synchronization issues that result in packet drops. It is assumed that each SN is assigned certain V_{BP} value when it is initially deployed. The nodes with high configuration are assigned more V_{BP} than the other nodes because these nodes are playing the role of CHs. The changes in V_{BP} are used for performing dynamic clustering as well as dynamic key generation during intra-cluster communication.

After the nodes are deployed in the network, the SNs pass through several functional states such as node-stay-alive, packet transmission as well as reception, encoding and decoding. During these states, the SN will forward other sensor's data or injecting its own data into the network. Based on the actions performed by the nodes, the associated powers are given in Table II.

When a source node detects any event, it forwards the packet size (Z) towards the sink. The V_{cs} is computed using the following equation:

$$V_{cs}=Z\cdot(P_{tx}+P_{enc}+P_{MAC})+t\cdot P_a+P_{SYNC}. \quad (1)$$

where Z = packet size, t = duration of alive state of the node.

When a CH receives the data from its member node, the V_{BP} can be updated by decrementing the cost associated with the actions performed by the sender. The V_{ci} is computed using the following equation.

$$V_{ci}=Z\cdot(P_{rx}+P_{auth}+P_{dec}+P_{tx}+P_{enc})+t\cdot 2\cdot P_a \quad (2)$$

Thus, the transient value of the V_{BP} is obtained by decrementing the previous V_{BP} (P_i), which is represented using the following equation.

$$V_{BP} = \begin{cases} P_i - V_{cs};\ if\ the\ packet\ received\ from\ Sender \\ P_i - V_{ci};\ if\ the\ packet\ is\ received\ from\ intermediate\ node \end{cases} \quad (3)$$

where P_i = previous V_{BP}

After every action, each node computes and updates the transient value of V_{BP}.

3.2.2 Cluster formation and CH election

Since the network consists of high configuration nodes, these nodes are usually selected as CHs. During the life time of the network, if any CH with high configuration has been compromised or dead, the nodes that are associated with that CH will be isolated from the network. To avoid such situation, the member nodes try to associate with another CH. If no other CH with high configuration is available in their nearby region, one among the normal nodes with high V_{BP} and less S_{av} has to take the responsibility as CH in order to extend the lifetime of the network. Hence, the CHs are selected based on the weight value. The weight values are not predetermined, instead they are dynamically calculated. When the nodes are deployed in the network, they estimate their weight value based on the parameters such as *ND*, D_{av}, S_{av} and V_{BP} [37] [38] [39]. The steps for cluster formation and maintenance are as follows:

Let *S* be the sink node, $\mathcal{N}$ be the number of sensor nodes in the WSN, each normal node is denoted as N_i and the high configuration nodes enabled with GPS are denoted as CH_i where $i \in \mathcal{N}$.

Step 1: All the nodes in the network broadcast a HELLO message to its neighbours. The nodes that are enabled with GPS only include its location information in the HELLO message. The other normal nodes includes (0, 0) as its location information. The nodes that receive the location information from the GPS enabled node, calculates its own *NL* based on DRL scheme [40]. The CH is considered as seed node for using in DRL scheme. This *NL* is updated by all N_i every time if they move to other location using the CH available in the target location.

Step 2: All the nodes including the GPS enabled nodes calculate its S_{av} [41] using the Equation (4) which is given below:

$$S_{av} = \frac{\sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2}}{\Delta t} \quad (4)$$

where (x_t, y_t) is the coordinates of node N_i at time *t*

(x_{t-1}, y_{t-1}) is the coordinates of node N_i at time *t-1*

Δt is the time interval among *t* and *t-1*

Step 3: All the nodes that receives the HELLO message calculates the D_{av} value using the Equation (5) which is given below:

$$D_{av} = \frac{\left(\sum \frac{\xi\sqrt{P_m/P_r}}{R}\right)}{No.\ of\ HELLO\ messages\ received} \quad (5)$$

where P_m and P_r are the transmit and receive powers, respectively, *R* is the transmission range and ε be the path loss exponent.

Step 4: Based on the received HELLO messages, all the N_i calculates the node degree [42] ND_i using the number of HELLO messages received from different one hop neighbours as:

$$ND_i = COUNT(N_i | D(N_i, N_j) < R_{tx}(i),\ i\ and\ j \in \mathcal{N}) \quad (6)$$

where $D(N_i, N_j)$ denotes the distance between N_i and N_j, $R_{tx}(i)$ denotes the transmission range of N_i.

Step 5: Once all the values that are required for CH election is calculated by every nodes in the network, they broadcast the CH_NOM message which consists of *ND*, D_{av}, S_{av}, V_{BP} and W_i as 0.

Step 6: Upon receiving the CH_NOM messages, all the nodes calculate the W_i as follows:

$$W_i = \frac{(c_1 D_{av}) \times (c_2(S_{av} + 1))}{(c_3 ND) \times (c_4 V_{BP})} \quad (7)$$

where $i = 1, 2, 3, ..., \mathcal{N}$ nodes, c_1,c_2,c_3 and c_4 are the weighted coefficients. The values of weighted coefficients are assigned in such a way that $c_1+c_2+c_3+c_4=1$. We have assigned the values for c_1, c_2, c_3 and c_4 as 0.15, 0.35, 0.15 and 0.35 respectively. During the normal case, preferably the node enabled with GPS is selected as CH. In some case, if no node that is enabled with GPS is available in a region, preferably the node with less mobility and higher V_{BP} has to be elected as CH. The later case happens if the CH is compromised or physically damaged. Based on the Equation (7), preferably the numerator values should be less and the denominator values should be

high for a node to become a CH.

Step 7: After estimating the weights of each N_i prior to the clustering set up, the N_i with minimum W_i is chosen as the CH. So all the N_i broadcasts the calculated W_i with in its transmission range using NOD_WGT message which consists of the information such as W_i and *ts*. Based on this information, the node with minimum W_i consider it as the CH in that region and broadcasts the CH_INFO message to its member nodes which consists of the information such as ID of the CH_i, *ND* and *ts*. Consider a set of five nodes in a region for which the sample values are given in the Table III. Among the five nodes, the weight value of the node number 10 is less. So the node number 10 will be selected as CH in that region. Similar operation is performed in all the other regions and the CHs are selected.

Step 8: Based on the received CH_INFO message and its signal strength, the N_i sends a CL_JOIN message to its nearest CH_i to join in the cluster. If a node N_i is located in an overlapping region, it may receive more than one CH_INFO messages. In that situation, the N_i selects a CH_i with good signal strength as its CH and sends a CL_JOIN message.

Step 9: The steps 1 to 8 have to be repeated if any CH_i moves to other location. Otherwise if a member node moves to other location, the *ND* value itself is updated and informed to the members of CH_i.

3.3 Dynamic key management

3.3.1 Cost Function

In the mobile environment, there are more possibilities for the attackers to be included in the network. The attackers will try to compromise the keys that are used by the legal nodes for security related activities. So the keys should be dynamically generated and used for security related activities. The K_D is generated by the source node and the same key is to be generated by the destination node also without exchanging any keying information. The K_D is further used for securing the data as well as for authentication. The key is generated dynamically by using one or more dynamic parameters that are identified by both the source and the destination nodes inside the network. We have identified the dynamic parameters such as *NL, ND* and $V_{BP.}$ Among these, the V_{BP} plays a major role. The CH maintains the details of each member ID along with its *NL, ND* and V_{BP} in its member table. The CH determines CCV for each SN based on these parameters which is also maintained in the member table.

In the proposed architecture, the location is estimated based on DRL [40]. The *NL* of each node is given by the (x, y) coordinates. The *ND* is estimated based on the neighboring nodes information. As *NL*, *ND* of a node are dynamic due to mobility, both the parameters changes dynamically. V_{BP} is dynamic in nature since it varies depending on the node state. The computation of cost function is as follows:

$$CCV = \alpha\,(xa + yb) + \beta\, ND + \gamma\, V_{BP}. \quad (8)$$

where α, β and γ are normalization factors and a, b are constants. The values of the normalization constants α, β and γ are chosen between 0 and 1 such that $\alpha+\beta+\gamma=1$. We have assigned the values for α, β, γ, a and b as 0.2, 0.2, 0.6, 0.5 and 0.5 respectively. Whenever the V_{BP} is decremented, the cost function is updated.

3.3.2 Dynamic key generation and encoding

The K_D is generated based on the transient values of the *CCV as* $K_D \leftarrow f(CCV)$. When a node N_i has data to transmit, it uses its current V_{BP} for generating the K_D. Using the K_D, the N_i encrypts the data and transmits the same to its associated CH. When the CH receives the data packet, it has to generate the same K_D used by N_i for decrypting the data. Using the size of

Table III *Sample Weight calculation*

Node Id	ND	D_{av}	S_{av}	V_{BP}	W_i
11	5	2.6	2	33.2	0.107
12	3	3.733	3	32.22	0.317
13	3	3.7	3	33.1	0.306
8	3	3.6	3	32.02	0.307
10	5	3.7	0	388.57	0.008

the received data and the previous V_{BP} of the node N_i, the CH generates the correct K_D. The encoding mechanism refers to the RC5 encryption mechanism. The key to RC5 is generated dynamically. The packets comprise the fields ID, type and data fields together denoted as *[z]*. Each SN forwards this packet to its CH. As per the result of the RC5 mechanism, *[z]* is transmitted in pseudorandom manner. The packet to be forwarded is included with *[ID {k (z)}]* and *k (z)* constitutes the message *z* encoded by key *k*. When the next CH along the path to sink receives the packet, it generates the K_D locally to decode the packet.

After the event detection, the source node needs to maintain the secured reports and it utilizes the V_{BP} to construct the next key. Further, the key is given as input to the RC5 algorithm inside the encoding module to generate a permutation code for encoding the *[z]* message.

3.3.3 Key updating technique

The key updating process occurs during the three cases which are N_i leaves a cluster, N_i joins a cluster and *CH* leaves the cluster. The steps involved in these three cases are shown in the Algorithm I.

Algorithm I Key Updating Process

if N_i leaves a cluster **then**
  N_i notifies *CH*
  CH authorizes N_i
  CH re-computes *CCV* of its members
else if N_i joins a cluster **then**
  N_i sends join request to *CH*
  CH sends acknowledgment to N_i
  N_i informs *CH* of its V_{BP}, *NL* and *ND*
  CH authenticates N_i
else if *CH* leaves a cluster **then**
  CH notifies its cluster members
  Re-election of *CH* takes place
  New *CH* re-computes *CCV* of its cluster members
end if

Algorithm II Authentication Procedure

N_i computes $HMAC = H(K_D, K_D(z))$
N_i sends *HMAC* to *CH* along with payload
CH derives K_D and compute $HMAC_{ch}$
Verify $(HMAC = HMAC_{ch})$
if valid **then**
  $M = D(K_D, K_D(z))$
  Aggregate & forward *M to D*
else
  message unauthenticated
end if

3.3.4 Authentication technique

The attackers try to compromise the keys that are used by the sensor nodes for authentication. If the attackers compromise the keys, they may disuse the authenticated nodes from the network. To prevent the attackers from cheating the authenticated nodes, the keys are dynamically changed and the same keys are used for generating HMAC code. By this the attackers are not able to use the previous keys for cheating the authenticated nodes. The steps involved in the authentication technique are shown in algorithm II.

3.4 Bi-directional malicious node detection

A major issue in the WSN is that the nodes may be compromised and used as malicious nodes to disrupt the network service. In the proposed scheme, a bidirectional malicious node detection mechanism has been included to improve the security. This mechanism identifies the malicious nodes among the CHs and the member nodes. The malicious member nodes are identified by the CHs and the malicious CHs are identified by the BS. To identify such malicious nodes, we define a trust based malicious node detection model based on Bao's [43] hierarchical trust management protocol.

3.4.1 Trust estimation

For trust estimation, we have considered three components such as direct interactions (*di*), number of transmission attempts (*ta*) and truthfulness (*tn*). The component *di* is used to identify the attacker nodes with high

power which try to communicate with nodes in longer distance. The component *ta* is used to identify very few or very high number of transmission attempts. The component *tn* is used to identify if a node pretends to be another node. Bao's model evaluates the trust based on the four components direct intimacy, honesty, residual energy and unselfishness. But we consider only three components that are adequate to build the trust value and identify many attacks launched by malicious nodes by reducing the overhead at CHs. The CH evaluates the trust value $T_{CHj}(t)$ of node j at time t, where $T_{CHj}(t)$ is between [0,1]. $T_{CHj}(t)$ is computed as:

$$T_{CHj}(t) = r_1 T^{di}_{CHj}(t) + r_2 T^{ta}_{CHj}(t) + r_3 T^{tn}_{CHj}(t) \quad (9)$$

where r_1, r_2 and r_3 are weights related to these trust components such that $r_1+r_2+r_3=1$. The trust component $T^{di}_{CHj}(t)$ is computed by the number of direct interactions of the node j which is having high power to communicate with CH in longer distance over the time period [0, t]. The trust component is computed by the number of transmission attempts which the node j has made with CH over the time period [0, t]. The trust component $T^{tn}_{CHj}(t)$ represents the truthfulness of node j measured by the CH if any node pretends to be another node. This can be identified by verifying the packet's source ID. If a node pretends to be another node, it uses the ID of another existing legal node in the network. To identify this, the CHs share the list of member nodes periodically that are associated with them.

When a node i evaluates a node j at time t, they update $T^{X}_{ij}(t)$ where X indicate a trust component as follows:

$$T^{X}_{ij}(t) = (1-\propto) T^{X}_{ij}(t-\Delta t) + \propto T^{X,direct}_{ij}(t) \quad (10)$$

where Δt is the trust update interval for the node j and is a weighing parameter for the trust values which represents the trust degradation over a time period. The node i use its new trust based on the direct observations which is calculated as:

$$T^{X,direct}_{ij}(t) = \sum_{l=1}^{t} T_{direct}(l) \quad (11)$$

where $T_{direct}(l)$ represents the trust value of j evaluated by i based on the direct observations at interval l. The value of T_{ij} is based on the status of node j. The values for T_{ij} is assigned as 0, 0.5 and 1 respectively if the status is completely malicious, ignorance and completely trustable. The node i is either BS or CH if node j is CH or cluster member respectively.

3.4.2 Malicious node detection algorithms

The following two algorithms illustrate the malicious node detection mechanism:

Malicious node detection

1. Initially each CH sets a TT_{CH} with respect to its sensing zone.

2. CH monitors the trust value of data transmitted by its CM_i and detects whether CM_i is malicious or not by using the following two cases:

Case 1: if $T_{CHj}(t) < TT_{CH}$, then CM_i will be marked as ML_i.

Case 2: if $T_{CHj}(t) > TT_{CH}$, then CM_i will be considered as trusted node.

When the trust value of the data transmitted by the cluster member is less than this threshold, the member node is considered to be malicious. Otherwise, it is considered as trusted node.

3. Following this malicious node detection, the CH broadcasts the malicious node *REVMES* to its members. The message includes the identity of ML_i and the K_D stored in it.

4. CM_i on receiving *REVMES* disconnects its link with ML_i and performs key refreshment associated with that node.

Using this phase, the malicious nodes are eliminated and the keys known by them are ejected from the network. Though the malicious cluster member nodes are eliminated from the cluster, there is a possibility for the CH to become a malicious node. The following phase is executed to validate the CHs in the network.

Malicious CH detection

1. BS computes $T_{BSj}(t)$ of CH and is verified with TT_{BS} using the following cases:

Case 1: if $T_{BSj}(t) > TT_{BS}$, then CH will be considered as trusted node.

Case 2: if $T_{BSj}(t) < TT_{BS}$, then CH is marked as malicious node.

2. Following the detection of malicious CH, BS broadcasts the CHIC to all the cluster members. The invalidation command transmitted by the BS is recognized only by CM_i and is revealed by CH_i, since BS and CM_i are protected using symmetric dynamic key.

3. CM_i upon receiving CHIC, terminates the data transmission to the malicious CH and performs the CH re-election.

3.4.3 Security analysis

The Bi-directional malicious node detection is used to identify and isolate the malicious nodes inside the network. We have used three trust components that can identify various attacks launched by the malicious nodes.

The trust component $T_{CHj}^{di}(t)$ identifies the Laptop class attacks, wormhole attacks and rushing attacks. These three attackers normally use long transmission radio transceivers that are identified by this component.

The trust component identifies the black-hole or grayhole attacks, packet dropping attacks, HELLO flood attacks, DoS attack and selective forwarding attacks. This component identifies the above mentioned attackers as:

Black hole or Gray hole attack & packet dropping attack: These attacks normally attract the routing paths to the destination through the attacker node, thereby create black hole and drop the packets. During such attack, it can be identified that more numbers of packets are received by the attacker and very minimum numbers of packets are transmitted from the attacker.

HELLO flood attack or DoS attack: The attacker transmits more number of HELLO packets or DoS packets that can be identified by this component.

Selective forwarding attack: The node selectively forwards the packets that are transmitted via the attacker nodes. In such case, the number of outgoing packets from the attacker will not be same as the average number of packets transmitted by normal nodes. Using this component, this attack will be identified.

The trust component identifies the Sybil attack based on the member nodes ID list maintained by CH. This component can detect Sybil attack if the attacker has stolen either the regular node's ID or a new fabricated node's ID.

IV. Performance Evaluations

4.1 Network threat model

In our Threat model, both the outside and inside attackers are considered. In case of outside attack, any malicious node can be placed inside the network and it can inject false data into network, selectively forwards the packet inside the network, and create black hole and wormhole attacks. In case of inside attacks the intruder compromises the keys by physically capturing the node, takes the control over it, launches replay attack, and injects false data into the network.

Our goal is to implement the malicious nodes detection technique to avoid inside attacks, which have the correct keys that want to become CHs, as well as to prevent the outside attacks. For this, the bidirectional malicious node detection technique is applied to identify outside and inside attackers among CMs and CHs. When the cluster member tends to join another cluster, the CH employs the authentication mechanism to avoid the inside attacks.

4.2 Simulation settings

The proposed technique CSDKT is evaluated through NS2 version 2.32 simulator. The simulation is performed over 20 test runs by varying the scenario and the average is taken for each value. The results are plotted as error graph with 95%confidence interval. The simulation parameters are shown in the Table IV.

The performance of CSDKT is compared with the LOCK [26] and the Authentication and Key Establishment (AKE) [34] schemes, since the LOCK uses similar clustered architecture as CSDKT and the AKE uses dynamic keying and supports mobility similar to CSDKT.

Table IV *Summarization of the simulation parameters*

Parameters	Values
No. of Nodes (n)	100 to 500
Area Size	500 m × 500 m
Simulation Time (sec)	300
Traffic Source	CBR
Mobility speed (m/s)	0 to 25
Initial Energy(J)	CH =50, SN =5
Initial V_{BP} (J)	CH =500, SN =50
Radio range (m)	CH =150, SN = 50
No. of Inside Attackers	90% of attackers
No. of Outside Attackers	10% of attackers
No. of CHs	6% of n

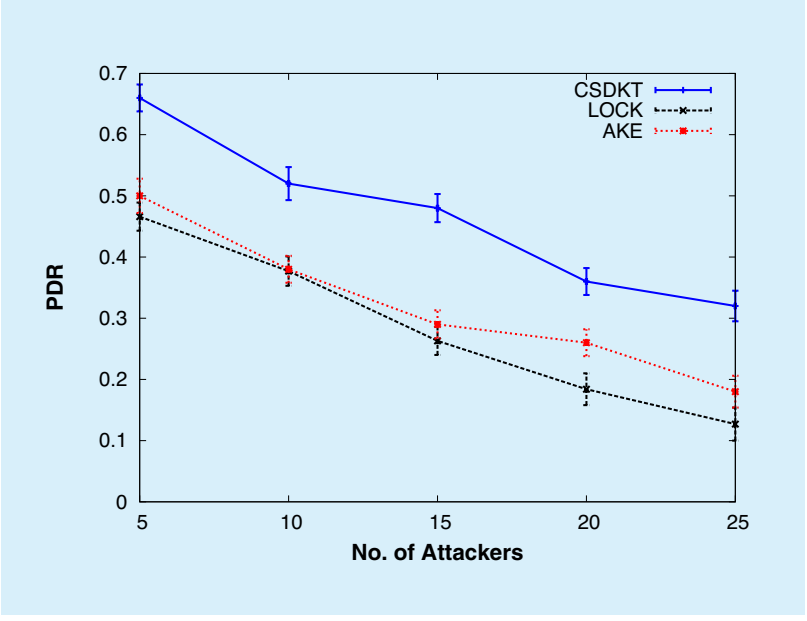

Fig.2 *Packet Delivery Ratio*

4.3 Simulation results

Keeping the number of nodes as 200, the number of attackers is varied as 5, 10, 15, 20 and 25. Firstly the average Packet Delivery Ratio (PDR) is measured. Fig. 2 shows the PDR of CSDKT, AKE and LOCK schemes.

The CSDKT eliminates both the malicious cluster member and the CH using bi-directional malicious node detection technique. Since the malicious nodes are eliminated from the network, the number of false packets injected inside the network is decreased and thereby more number of legal packets reaches the destination. Also CSDKT uses multi-path dispersal routing technique which mitigates the selective forwarding attacks. Hence CSDKT has shown better in PDR, when compared to LOCK and AKE.

Secondly the average energy consumed for the data transmission by all the nodes is measured. It includes energy spend for sending, receiving and computing. Fig. 3 shows the average energy consumption of CSDKT, AKE and LOCK schemes. The average energy consumption increases, when the number of attackers is increased. This is due to the number of false packets increased by the attackers in the network. The CSDKT filters the bad packets at the CH level itself. The packets from the attackers do not propagate throughout the network and results in less energy consumption. In LOCK, if the CH node is captured, it has to initiate re-keying. The new CH is elected by BS. After a new CH is elected with the help of Key Generation Node (KGN), the CH distributes the new keys to its members. This consumes more amount of energy. The AKE re-initiates the key establishment procedure on expiry of key life-time and the energy consumption will be high. In AKE, when a node joins in the network, it sends the requirement message to the BS. This message is sent through the foreign cluster which requires more communication overhead. This communication overhead incurs more energy in the network. Since the CHs are not capable of recognizing any compromised nodes in time, the sensor nodes have to reset its sub-station IDs and re-establish keys with its nearer CHs via BS. This also consumes more energy.

Thirdly, the effect of network resilience has been analyzed by the percentage of success the network of 500 nodes withstands in the presence of 25 numbers of attacks. Figs. 4a, 4b and 4c show three attacking scenarios respectively. CSDKT detects the malicious nodes and excludes them from the network. Further the malicious nodes do not participate in the network events. Even the nodes are compromised by the attackers; it cannot af-

fect the whole network because the key to the next session dynamically changes based on the changes in CCV components. As a result it will be difficult for the attackers to perform node capturing attack.

Fig. 4a shows the success rate of the network in the presence of 25 stationary attackers. In CSDKT, the percentage of alive nodes reaches around 88% and becomes stable because all the malicious nodes are identified and isolated from the network.

In LOCK, the re-keying is initiated only if the node capture comes nearer to Network Resilience Point (Nc). Until re-keying, the nodes are compromised in the network. Also the attackers use the compromised nodes to further compromise the remaining nodes in the network. Due to this, the percentage of alive nodes is decreasing quickly.

In AKE, there is a chance that the key-cache or the pre-distributed cluster key stored in each node can be captured. Once these keys are captured, the attackers further tries to compromise more number of nodes until AKE re-initiates re-keying. Since AKE supports mobility, the neighbor node of any attacker node moves to another cluster and thereby it performs better than LOCK.

Fig. 4b shows the success rate of the network in the presence of 25 mobile attackers. The mobility speeds of the attackers are not same. So the attacker with maximum speed is able to attack most of the nodes by moving to another cluster quickly and launch attacks before the attackers are recognized and revoked from the network. The CSDKT performs well compared to the AKE and the LOCK because it identifies the malicious nodes and isolates from the network. Compared to stationary attackers, the CSDKT takes more time to identify the mobile attackers.

Fig. 4c shows the success rate of the network in the presence of 10 mobile and 15 static attackers. From the figure we can see that the static attackers are identified but before the network become stable, the mobile attackers are able to attack the nodes by moving to another cluster and launching attacks. Since the

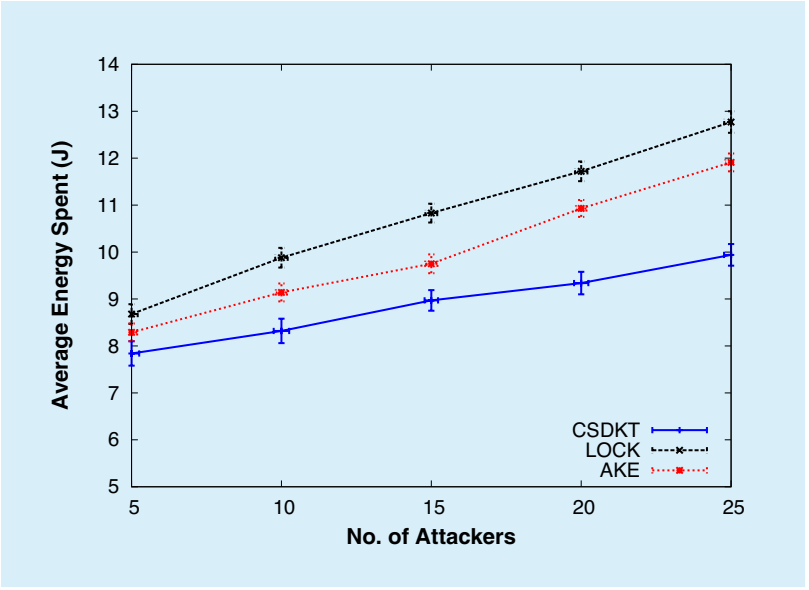

Fig.3 *Average Energy Consumption*

CSDKT and the AKE support mobility in the network, it performs better than the LOCK. Due to the dynamic keying CSDKT, the attackers are not able to compromise the other nodes based on the keying information available in a node. Hence the CSDKT performs well compared to AKE and LOCK

We evaluate the FCC by varying the attackers and network size. In this analysis, the network size is varied from 100 to 400; the attackers are varied from 5 to 25%. The obtained results are presented in Fig.5. We can clearly see in Fig. 5 that the FCC is increasing quickly in LOCK and AKE schemes when the number of nodes and percentage of attackers are increased. In CSDKT, when the percentage of attackers are increased, the FCC also increases quickly when n=100 and little bit lesser when n=200. But, when n=300 and 400, the FCC for CSDKT is almost stable. It is because the CSDKT identifies the malicious nodes as well as malicious CHs and eliminates them from the network. Also the CSDKT uses multi-path dispersal routing where packets are routed through multiple paths and there by it mitigates the attackers. In LOCK and AKE, all the packets from a source are routed in the same path to the destination. If more number of attackers is placed in that path, the packets are

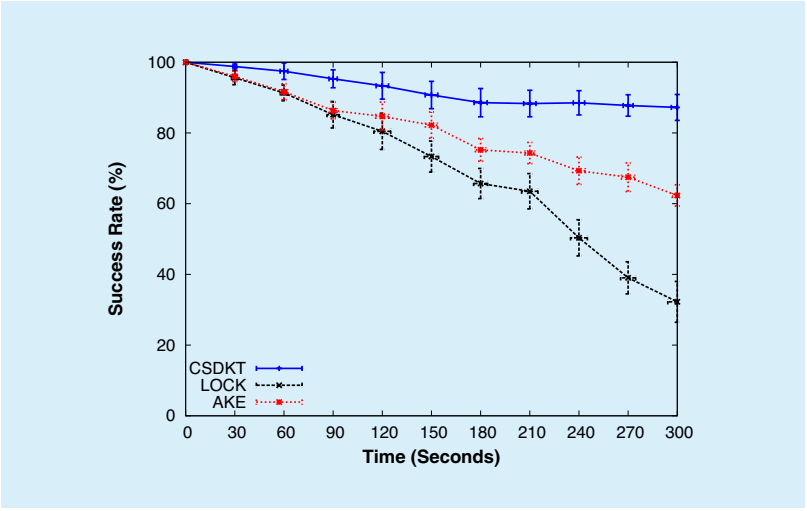

Fig. 4*a Network Success Rate with 25 Static Attackers*

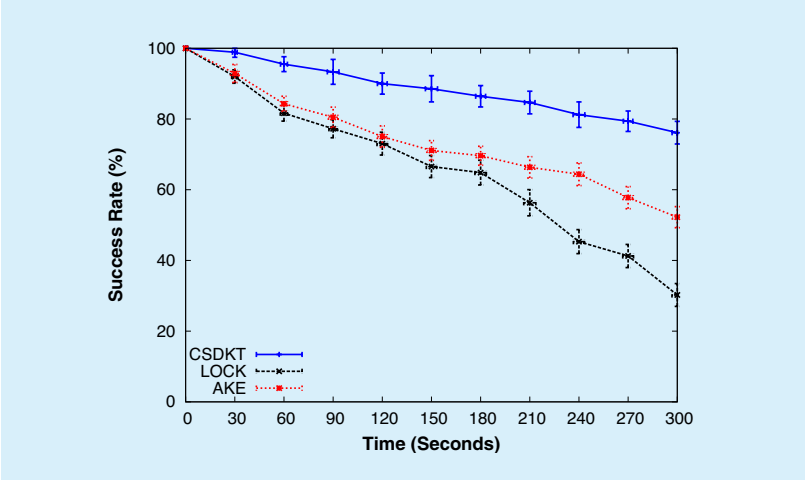

Fig. *4b Network Success Rate with 25 Mobile Attackers*

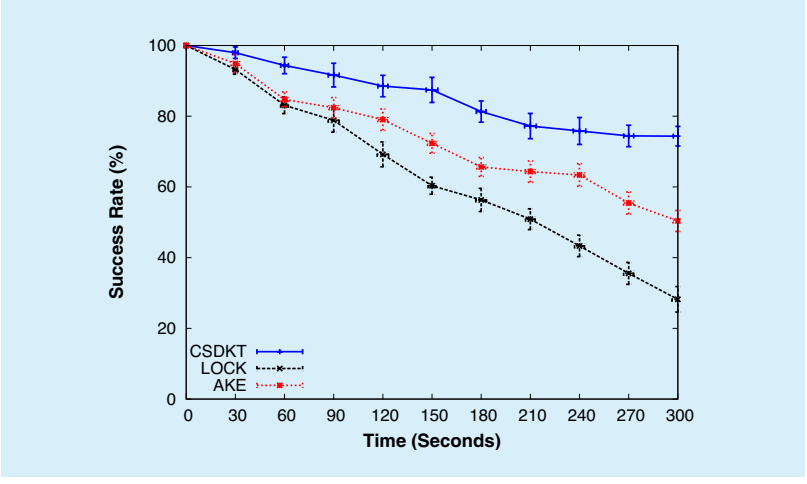

Fig. 4c *Network Success Rate with Static & Mobile Attackers*

compromised frequently. The AKE requires regularly that each SN to reset its sub-station ID to the real BS and to re-establish keys. Due to this, the CHs are not able to recognize the compromised nodes in time.

For the network that supports node mobility, the number of control packets to be exchanged between the nodes for the key establishment and handover will be increased. Hence, we have analyzed these overheads by keeping the number of nodes as 100 with 5% of attackers and the node speed is varied from 5m/s to 25m/s. The control overhead occurs due the node mobility which is measured in terms of number of control packets exchanged over the total number of packets. The result is given in Fig.6. It shows that the CSDKT has the lesser overhead compared to AKE, when the speed is increased because the CSDKT does not require re-keying during node mobility. Due to high mobility, the LOCK has to send more number of control messages frequently for re-keying.

V. Conclusion

In this paper, a secure mobility aware dynamic keying technique for authentication in WSN is proposed. In this scheme, the CH is chosen based on the weight value which is estimated using parameters such as the *ND*, D_{av}, S_{av}, and V_{BP}. A bidirectional malicious node detection technique is employed and it eliminates the malicious nodes and the malicious CH from the network. When the cluster member tends to leave or join the network, the CH employs the authenticated key management mechanisms to ensure the security. When the CH tends to leave the network, the CH re-election is performed by the cluster member nodes. By simulation, it is proved that the proposed scheme prevents both insider and outsider attacks. By increasing the number of nodes, number of attackers and number of rounds, it is proved that the proposed scheme provides efficient security with reduced energy consumption, control overhead and packet loss. Also during high mobility, the proposed scheme performs well compared to AKE and

LOCK. In this work, if any malicious node overhears the V_{BP} or CCV values, there is a chance of identifying the dynamic keys.

References

[1] Patrick Traynor et al., "Efficient hybrid security mechanisms for heterogeneous sensor networks," IEEE Transactions on Mobile Computing, vol. 6, pp. 663-677, 2007.

[2] Yun Wang, Xiaodong Wang, Bin Xie, Demin Wang, and Dharma P Agrawal, "Intrusion detection in homogeneous and heterogeneous wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 7, no. 6, pp. 698–711, 2008.

[3] Ossama Younis and Sonia 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.

[4] Dilip Kumar, Trilok C. Aseri, and R. B. Patel, "EEHC: Energy efficient heterogeneous clustered scheme for wireless sensor network," Computer Communications, vol. 32, no. 4, pp. 662–667, 2009.

[5] Elbhiri. B, Saadane. R, and Aboutajdine. D, "Stochastic Distributed Energy-Efficient Clustering (SDEEC) for Heterogeneous Wireless Sensor Networks," ICGST International Journal on Computer Network and Internet Research, CNIR, vol. 9, no. 2, pp. 11-17, 2009.

[6] Song MAO and Cheng-lin ZHAO, "Unequal clustering algorithm for WSN based on fuzzy logic and improved ACO," The Journal of China Universities of Posts and Telecommunications, vol. 18, no. 6, pp. 89-97, 2011.

[7] Dusit Niyato, Ekram Hossain, and Sergio Camorlinga, "Remote Patient monitoring service using heterogeneous wireless access networks: Architecture and optimization," IEEE Journal on Selected Areas in Communications, vol. 27, no. 4, pp. 412-423, 2009.

[8] Eleni Klaoudatou, Elisavet Konstantinou, and Georgios Kambourakis, "A survey on cluster-based group key agreement protocols for WSNs," IEEE Communications Surveys Tutorials, vol. 13, no. 3, pp. 429– 442, 2011.

[9] Kirusnapillai Selvarajah, Carl Shooter, Luca Liotti, and Alan Tully, "Heterogeneous wireless sensor network for transportation system applications," International Journal of Vehicular Technology, vol. 853948, pp. 14 Pages, 2011.

[10] Zoran S. Bojkovic, Bojan M. Bakmaz, and Miodrag R. Bakmaz, "Security issues in wireless sensor networks," International Journal of Communications Letters, vol. 2, no. 1, p. 106–115, 2008.

[11] Aylin Aksu, Prashant Krishnamurthy, David Tipper, and Ozgur Ercetin, "On Security and Reliability Using Cooperative Transmissions in Sensor Networks," Mobile Networks and Applications, vol. 17, no. 4, pp. 526-542, 2012.

[12] Ashraf Wadaa, Stephan Olariu, Larry Wilson, and Mohamed Eltoweissy, "Scalable cryptographic key management in wireless sensor networks," in Int. Conf. on Distributed Computing Systems Workshops, 2004, pp. 796–802.

[13] Fei Hu, Waqaas Siddiqui, and Krishna Sankar, "Scalable Security in Wireless Sensor and Actuator Networks (WSANs): Integration Re-keying with Routing," Computer Networks, vol. 51, no. 1, pp. 285-308, 2007.

[14] Yixin Jiang, Chuang Lin, Minghui Shi, and Xuemin (Sherman) Shen, "Self-healing group key distribution with time-limited node revocation for wireless sensor networks," Ad Hoc Networks,

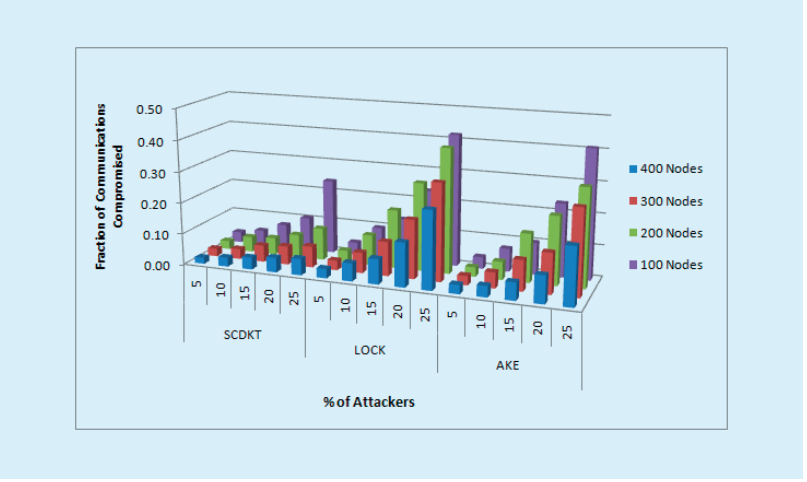

Fig.5 *Fraction of Communications Compromised*

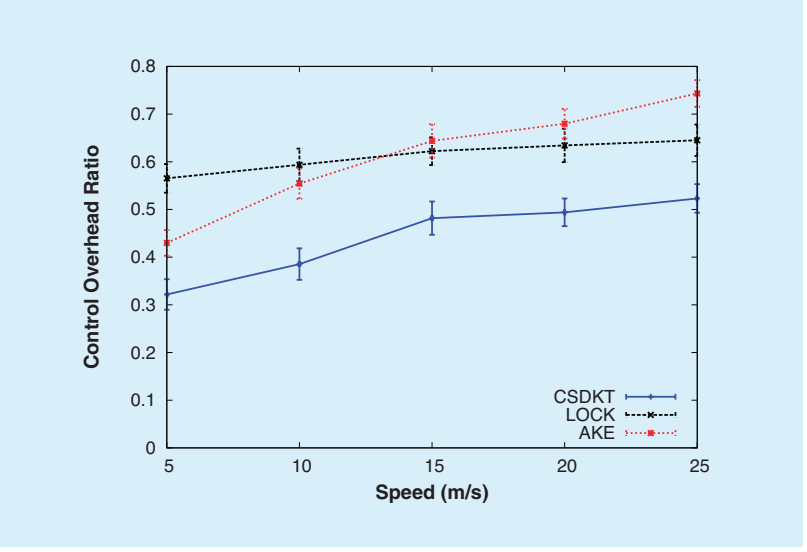

Fig.6 *Control Overhead by varying the speed of the Nodes*

vol. 5, no. 1, pp. 14-23, 2007.
[15] Wenliang Du, Jing Deng, Yunghsiang S. Han, and Pramod K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," ACM Transactions on Information and System Security (TISSEC), vol. 8, no. 2, pp. 228–258, 2005.
[16] Wenjun Gu, Dutta Neelanjana, Chellappan Sriram, and Xiaole Bai, "Providing end to end secure communication in wireless sensor networks," IEEE Transactions on Network and Service Management, vol. 8, no. 3, pp. 205-218, 2011.
[17] Sushmita Ruj, Amiya Nayak, and Ivan Stojmenovic, "Fully secure pairwise and triple key distribution in wireless sensor networks using combinatorial designs," in IEEE INFOCOM 2011, pp. 326–330.
[18] Xiaojiang Du, Mohsen Guizani, Yang Xiao, and Hsiao-Hwa Chen, "A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," IEEE Transactions on Wireless Communications, vol. 8, no. 3, pp. 1223– 1229, 2009.
[19] Yi Cheng and Dharma P Agrawal, "An improved key distribution mechanism for large-scale hierarchical wireless sensor networks," Ad Hoc Networks, vol. 5, no. 1, pp. 35–48, 2007.
[20] Bellare M, Canetti R Krawczyk H, "HMAC: Keyed-hashing for message authentication," 1997.
[21] Drake Patrick Mirembe and Maybin Muyeba, "Security issues in Ambulatory Wireless Sensor Networks (AWSN) Security Vs mobility," in 5th Annual Int. Conf. on Computing and ICT Research (CCIR'09), vol. 6167, Kampala, Uganda, 2009, pp. 289–301.
[22] Ronald L Rivest, "The RC5 Encryption Algorithm," in The Second International Workshop on Fast Software Encryption (FSE) 1994e, 1994, pp. 86-96.
[23] Thiruppathy Kesavan V and Radhakrishnan S, "Cluster based dynamic keying technique for authentication in wireless sensor networks," in Mobile Communication and Power Engineering.: Springer, 2013, vol. 296, pp. 1-8.
[24] Tzung-Pei Hong and Cheng-Hsi Wu, "An Improved Weighted Clustering Algorithm for Determination of Application Nodes in Heterogeneous Sensor Networks," Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, pp. 173-184, 2011.
[25] Heinzelman W. B, Chandrakasan A. P, and Balakrishnan H, "An application-specific protocol architecture for wireless microsensor networks," in Wireless Communications, IEEE Transactions on, vol. 1, 2002, pp. 660 - 670.
[26] Mohamed Eltoweissy, Mohammed Moharrum, and Ravi Mukkamala, "Dynamic key management in sensor networks," IEEE Communications Magazine, vol. 44, no. 4, pp. 122-130, 2006.
[27] Mohamed Eltoweissy, Ashraf Wadaa, Stephan Olariu, and Wilson, "Group key management scheme for large-scale sensor networks," Ad Hoc Networks, vol. 3, no. 5, pp. 668-688, 2005.
[28] Ozgur Koray Sahingoz, "Large scale wireless sensor networks with multi-level dynamic key management scheme," Journal of Systems Architecture, vol. 59, no. 9, pp. 801-807, 2013.
[29] Shu Yun Lim and Meng-Hui Lim, "Energy efficient and scalable group key management for hierarchical sensor network," Journal of Ubiquitous Systems & Pervasive Networks, vol. 2, no. 1, pp. 39-47, 2011.
[30] Qiuhua Wang, Huifang Chen, Lei Xie, and Kuang Wang, "One-way hash chain-based self-healing group key distribution scheme with collusion resistance capability in wireless sensor networks," Ad Hoc Networks, vol. 11, no. 8, pp. 2500-2511, 2013.
[31] Tim Landstra, Sarangapani Jagannathan, and Maciej J Zawodniok, "Energy efficient hybrid key management protocol for wireless sensor networks," in IEEE International Conference on Local Computer Networks, 2007, pp. 1009–1016.
[32] Biswajit Panja, Sanjay Madria, and Bharat Bhargava, "Energy and communication efficient group key management protocol for Hierarchical Sensor Networks," in IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, vol. 1, 2006, pp. 384–393.
[33] Pardeep Kumar, Mika Ylianttila, Andrei Gurtov, Sang-Gon Lee, and Hoon-Jae Lee, "An efficient and adaptive mutual authentication framework for heterogeneous wireless sensor network-based applications," Sensors (Base), vol. 14, no. 2, pp. 2732-55, February 2014.
[34] Ying Qiu, Jianying Zhou, Joonsang Baek, and Javier Lopez, "Authentication and key establishment in dynamic wireless sensor networks," Sensors, vol. 10, no. 4, pp. 3718-3731, 2010.
[35] Shun-Sheng Wang, Kuo-Qin Yan, Shu-Ching Wang, and Chia-Wei Liu, "An Integrated Intrusion Detection System for Cluster-based Wireless Sensor Networks," Expert Systems with Applications, vol. 38, no. 12, pp. 15234–15243, 2011.
[36] Xiao Zhenghong and Chen Zhigang, "A Secure Routing Protocol with Intrusion Detection for Clustering Wireless Sensor Networks," in in Information Technology and Applications (IFITA), 2010 International Forum on, vol. 1, 2010, pp. 16-18.
[37] Rico Radeke and Stefan Türk, "Node Degree based Improved Hop Count weighted Centroid Localization Algorithm," in Int. Conf. on Communication in Distributed Systems (KiVS'11), Kiel, Germany, 2011, pp. 194-199.

[38] Sung-Chan Choi, Seong-Lyong Gong, and Jang-Won Lee, "An average velocity-based routing protocol with low end-to-end delay for wireless sensor networks," IEEE Communications letters, vol. 13, no. 8, pp. 621–623, August 2009.
[39] Arif Selcuk Uluagac, Raheem A Beyah, Yingshu Li, and John A Copeland, "VEBEK: Virtual energy-based encryption and keying for wireless sensor networks," IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 994– 1007, 2010.
[40] Yi-Ling Hsieh and Kuochen Wang, "Efficient localization in mobile wireless sensor networks," in IEEE Int. Conf. on Sensor Networks, Ubiquitous, and Trustworthy Computing, (SUTC'06), vol. 1, 2006, pp. 292–297.
[41] Andrey Koucheryavy and Ahmed Salim, "Prediction-based Clustering Algorithm for Mobile Wireless Sensor Networks," in 12th International Conference on Advanced Communication Technology (ICACT), 2010, pp. 1209 – 1215.
[42] Sharad Saxena, Shailendra Mishra, and Mayank Singh, "Clustering Based on Node Density in Heterogeneous Under-Water Sensor Network," I. J. Information Technology and Computer Science, vol. 5, no. 7, pp. 49-55, 2013.
[43] Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee Cho, "Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection," IEEE Transactions on Network and Service Management, vol. 9, no. 2, pp. 169-183, June 2012.

Biographies

Thiruppathy Kesavan Venkatasamy, completed his M. E. and Ph. D in the field of Computer Science and Engineering from Anna University, Chennai, India, and Kalasalingam University, Krishnankoil, India respectively. He has more than 13 years of Teaching Experience from 2003 onwards. From 2016, he is working as Senior Assistant Professor in the department of CSE in Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India. His areas of interest include Wireless Sensor Networks, Computer Networks, System Software and Microprocessors. The corresponding author, E-Mail: vtkesavan@gmail.com

Radhakrishnan Shanmugasundaram *(Deceased),* completed his M. Tech. and Ph. D. in the field of Biomedical Engineering from Indian Institute of Technology. He has more than 25 years of Teaching and Research Experience. He worked as Senior Research Fellow at School of Biomedical Engineering, IT-BHU during 1988-1992. He served as Asst. Professor and Principal In-charge at Watumull Institute Electronic Engineering and Computer Technology (WIEECT), Mumbai during 1992-1996. From 1997, he was working in the Kalasalingam University (Previously known as Arulmigu Kalasalingam College of Engineering) in the Department of CSE. His fields of interest are Computer Networks, Bio-inspired computing and Computer Applications in Medicine. He has published 70 publications in various International journals.