CHAPTER-3

ZSDA: ZONE BASED SECURE DATA AGGREGATION SCHEME FOR CLUSTERED WIRELESS SENSOR NETWORKS

Security and energy efficiency are very critical in designing sensor networks. Energy efficiency is achieved with the help of data aggregation techniques whereas security is achieved by maintaining confidentiality between sensor node and base station. This Chapter presents a secure data aggregation scheme to address security and energy efficiency issues together.

Rest of the chapter is organized as follows:

Section 3.1 presents a system model for secure data aggregation scheme used for WSN. Key computation scheme have been described in Section 3.2. A lightweight encryption scheme is given in Section 3.3 followed by simulation results in Section 3.4. Finally, Chapter has been summarized in Section 3.5.

3.1.SYSTEM MODEL

To achieve energy efficiency, an efficient mechanism of cluster head selection is introduced to minimize the overlap area covered by one or more cluster heads. In existing data aggregation protocols, a single sample is calculated by applying an aggregated function on the reading of all the sensor nodes in a cluster. This operation is performed by the cluster head. There are two drawbacks of this scheme. The first drawback is that the amount of data received by the base station is very less because the reading of several cluster members is converted to a single reading and this single reading is received by the base station from each cluster which affects the overall results of the cluster. The second drawback of existing secure data aggregation schemes is that it breaks the principle of confidentiality between a sensor node and the base station because the actual reading of a sensor node is disclosed to the cluster head. So in presented data aggregation protocols, these problems have been addressed properly. In first case, instead of sending a single aggregated sample from cluster head to the base station, one sample from each duplicate class is transferred from a sensor node to the base station. This protocol also maintains the principle of confidentiality between a sensor node and the base station as the actual sensor reading is hidden from the cluster heads. Instead of sending the actual reading to the cluster head, the sensor node sent a pattern code to the cluster head. This pattern code is more than enough to compare the redundancy among the readings of two different sensors. This scheme also provides a security framework between sensor nodes and aggregator and in between sensor node and the base station. Simulation results prove that when the redundancy factor increases in the network, energy consumption decreases automatically and hence improves the performance of the system. This section describes the presented ZSDA (Zone based secure data aggregation) scheme for clustered wireless sensor networks. The system is divided mainly into three modules:

- Clustering
- Secure data aggregation
- Establish Security Capabilities

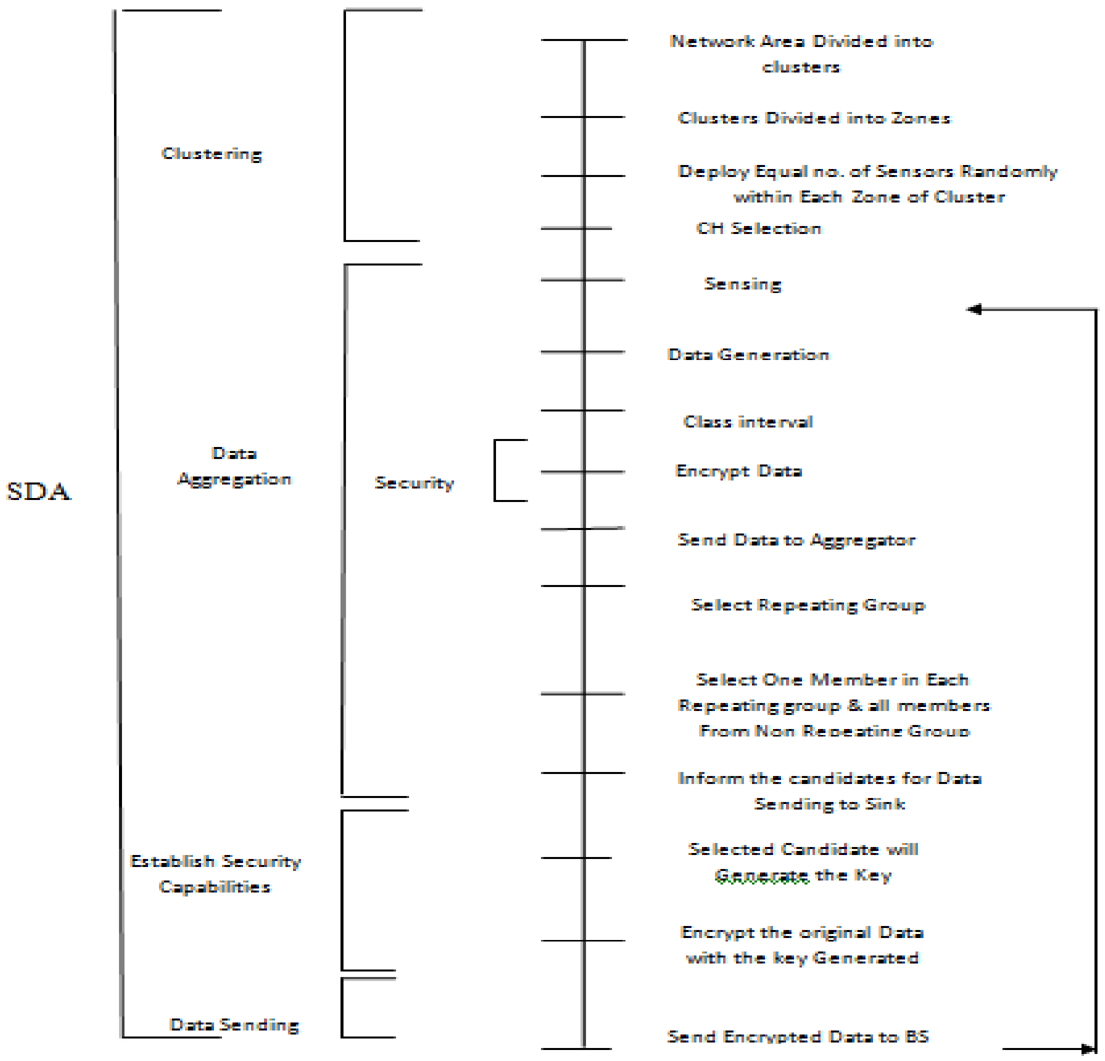
SDA
Clustering
Data Aggregation
Establish Security Capabilities
Data Sending
Security
Network Area Divided into clusters
Clusters Divided into Zones
Deploy Equal no. of Sensors Randomly within Each Zone of Cluster
CH Selection
Sensing
Data Generation
Class interval
Encrypt Data
Send Data to Aggregator
Select Repeating Group
Select One Member in Each Repeating group & all members From Non Repeating Group
Inform the candidates for Data Sending to Sink
Selected Candidate will Generate the Key
Encrypt the original Data with the key Generated
Send Encrypted Data to BS

Figure 3.1: System Model

3.1.1. Cluster Head Selection

Clustering is the process of creating groups of sensors based on common characteristics or criteria. All the sensor nodes in the network become the member of some group. The decision rule is applied to decide the group for sensor nodes. Individual sensor node is elected as a cluster head from each group. The role of cluster head is changed on rotation basis, i.e. cluster head in current round may not be elected as a cluster head in next round. There are many schemes that are used for cluster head selection. The schemes are given below:

3.1.1.1. Random Cluster Head Selection from Entire Network

In this scheme, all the cluster heads are elected randomly from the entire network some probability function as shown in the Figure 3.2.

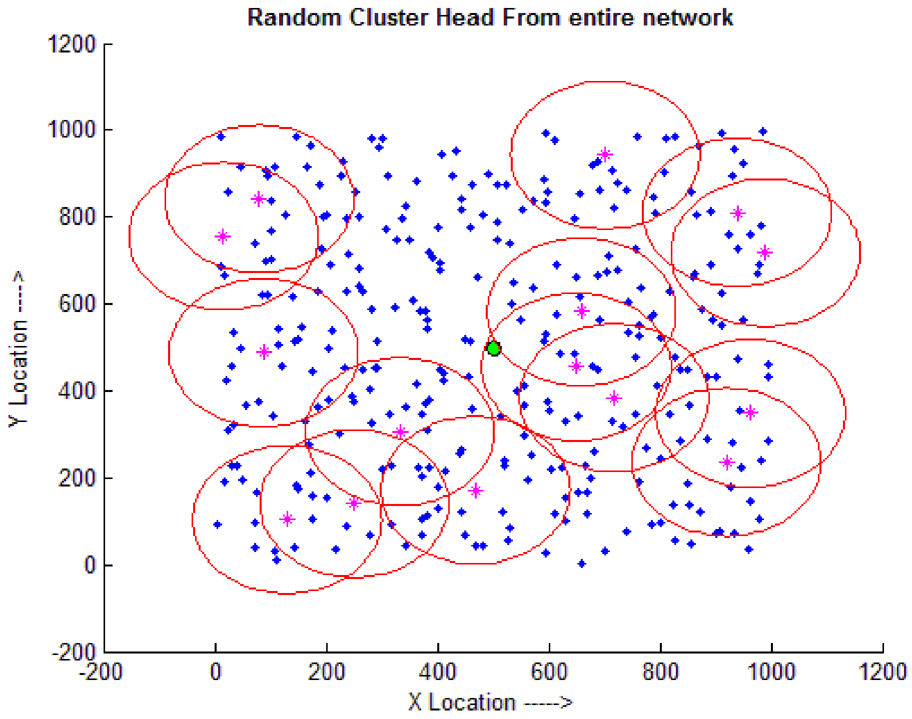

Figure 3.2: Selection of Random Cluster Head from Entire Network

Figure 3.2 show that there is a large overlapping area of the cluster head and many sensors not come in the range of any cluster head. Since energy dissipation in this type of network is very large.

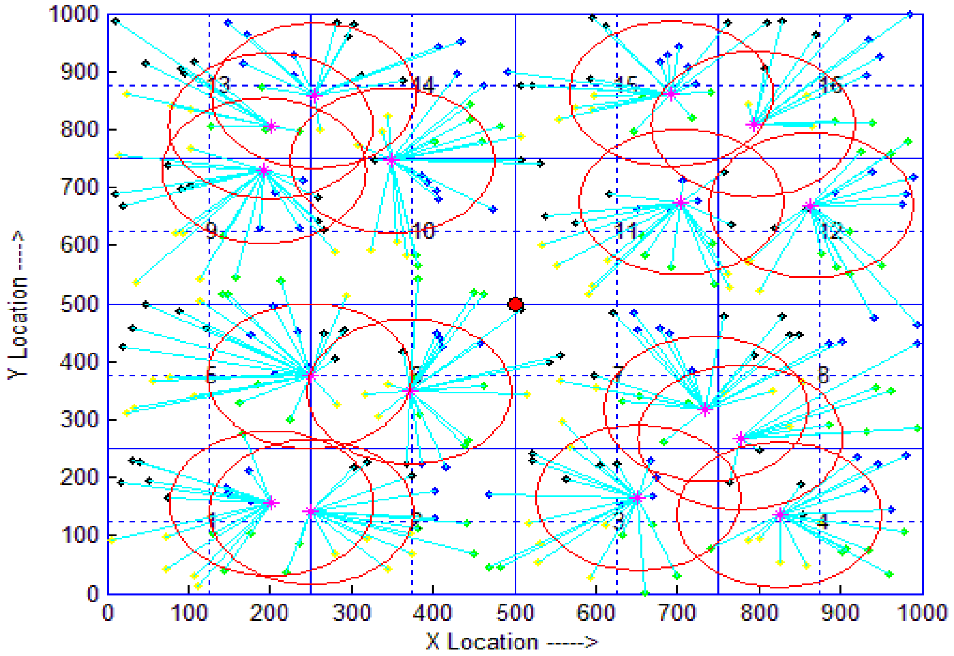

Figure 3.3: Selection of Random Cluster Head from Each Cluster

3.1.1.2. Random Cluster Head Selection from Every Cluster

In this technique the entire area is divided into clusters. The clusters are made by dividing the entire area into equal number of rows and columns. In this approach only one cluster head is elected from each cluster as illustrated in Figure 3.3.

In this approach it can be seen that the CH chooses is fixed in number. In this approach CH chosen from two neighboring clusters are very close to each other and hence there is no benefit of selecting two CH from this region. To overcome this problem a new clustering technique is introduced in presented model.

3.1.1.3. Presented Random Cluster Head Selection from Every Zone

Presented scheme uses the concept of physical clustering, i.e. every cluster is identified by its physical boundary. The boundary of each cluster is similar in size. Number of clusters into horizontal and vertical directions is decided in the setup phase. All the clusters are equal in size and each cluster is identified by its unique cluster ID. The area of every cluster is further subdivided into zones (four zones in each cluster, i.e. Bottom Left Zone (BLZ), Bottom Right Zone (BRZ), Upper Left Zone (ULZ), and Upper Right Zone (URZ). Now a cluster head is selected form same zone in all the clusters. The procedure is shown in Figure 3.4.

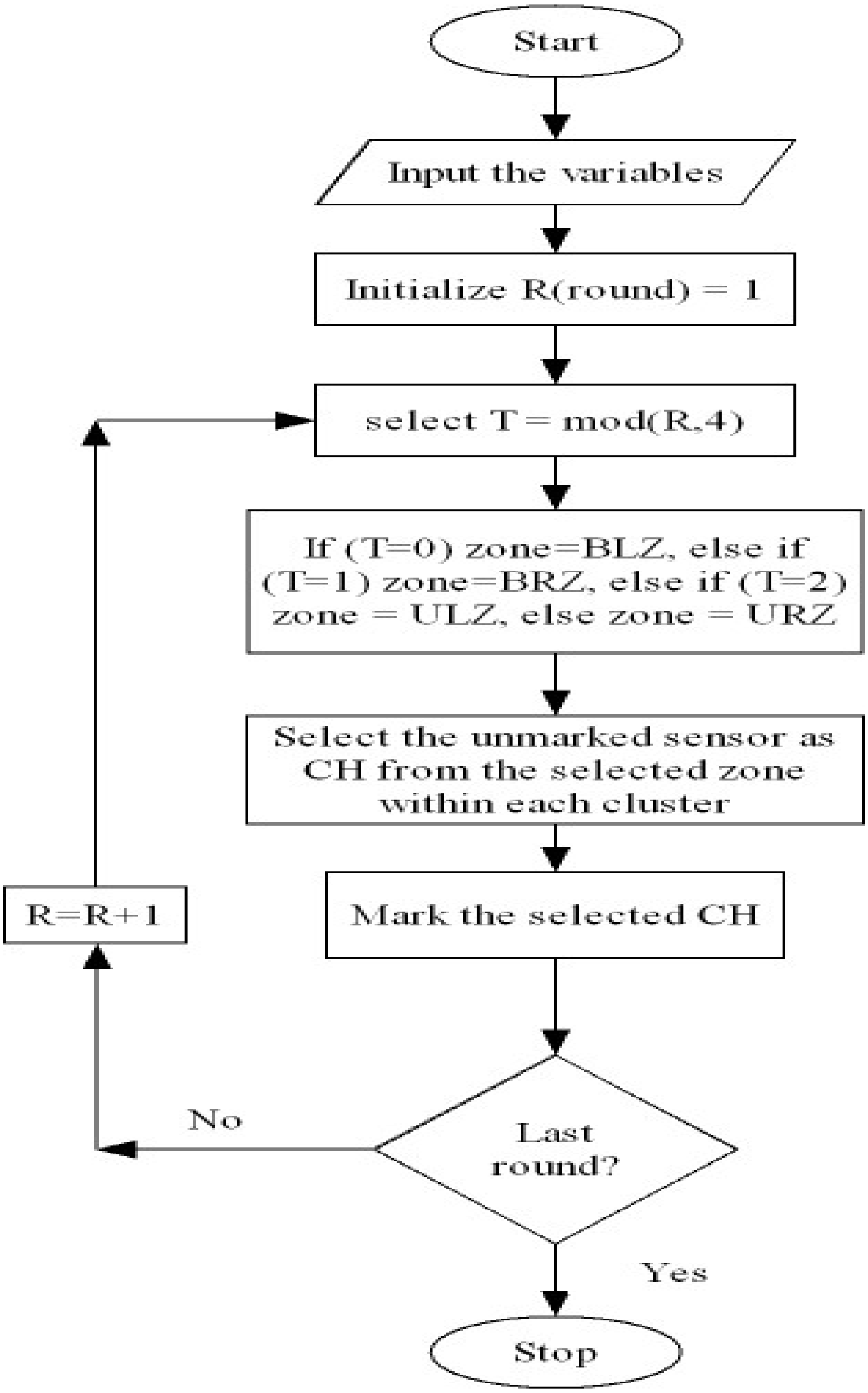
Start
Input the variables
Initialize R(round) = 1
select T = mod(R,4)
If (T=0) zone=BLZ, else if (T=1) zone=BRZ, else if (T=2) zone = ULZ, else zone = URZ
Select the unmarked sensor as CH from the selected zone within each cluster
Mark the selected CH
Last round?
No
R=R+1
Yes
Stop

Figure 3.4: Selection of Cluster Head from Each Cluster Area

The result of selecting cluster head from same zone in all the clusters is the large area covered by the cluster head, with minimum overlapping as shown in Figure 3.5. After each round, zone of each cluster is updated to select the new cluster head from this zone, i.e. zone is updated from BLZ to BRZ and from BRZ to ULZ and from ULZ to URZ and similarly from URZ to BLZ.

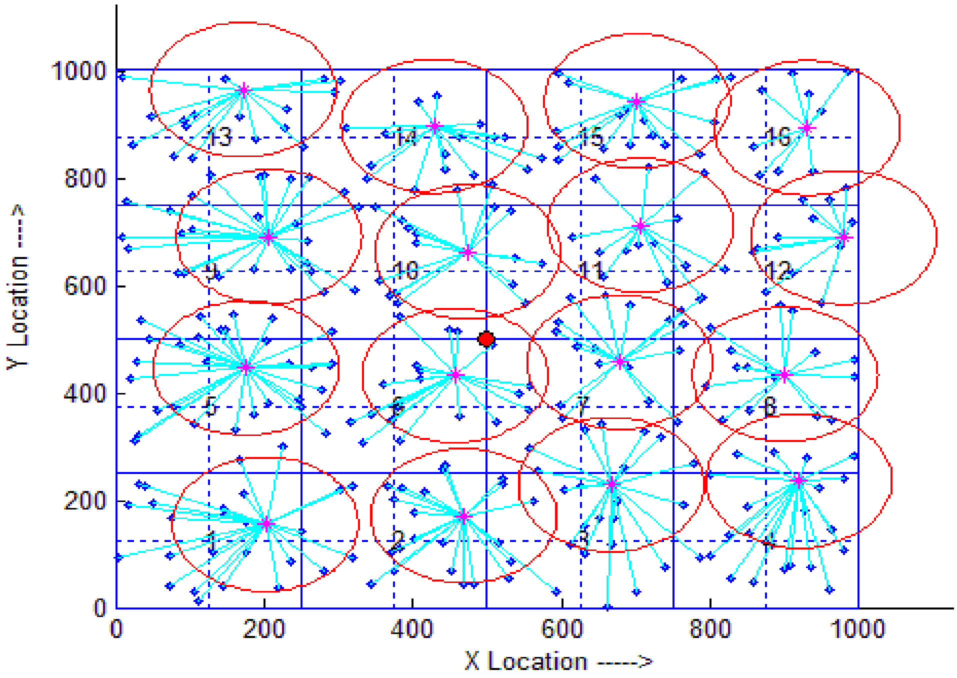

Figure 3.5: Random Cluster Head Selection from Same Zone within Each Cluster

3.1.2. Presented Secure Data Aggregation Scheme

Data aggregation is an effective technique for removing data redundancy and improving energy efficiency in WSNs. In direct data transmission scheme data is sent directly from sensors to the BS. In this scheme, redundant data is transferred across the network as shown in Figure 3.6 that consumes a huge amount of energy in the network. Presented scheme overcome this problem and ensures secure and energy efficient data aggregation.

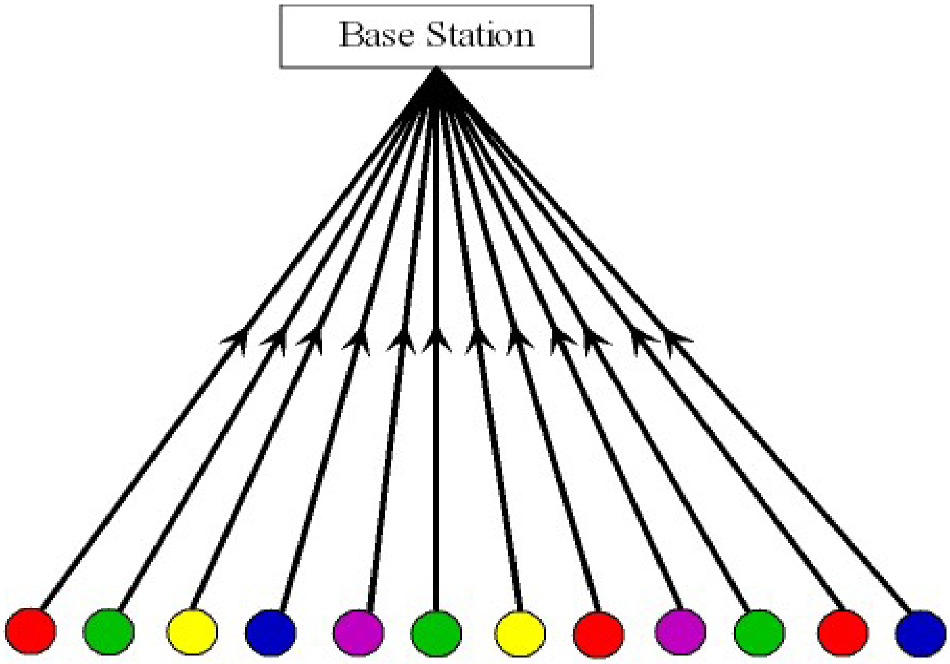

Figure 3.6: Data Transfer Scheme without Data Aggregation

The basic idea in presented scheme is to collect data received from different sensors and then compare this data to remove redundancy before submitting it to base station. Aggregator plays an important role in decision making. All the sensors sent their reading to the aggregator but for the security reason, actual reading or data is not sent to the aggregator.

According to data collected from the environment, various classes are used to define the categories. Now on the basis of sensor reading, every sensor selects corresponding value to the range interval. For this purpose a range value for each type of event is defined. These values are collected stored in the form of a table along with the event class interval and finally stored in the memory of each sensor. This table is known as mapping table. When a sensor senses the data, this data is searched in the mapping table for the corresponding range value to replace the event data value sensed by the sensor with the corresponding range value from the mapping table.

For example if a sensor within the network senses data between 1 to 50, in a network with 5 different classes stared from 1-10, 11-20, 21-30, 31-40, and 41-50. Sensors that have data in between this range falls into corresponding class, i.e. if the reading sensed by the sensor is 24 then it has been replaced by 6 and 6 is sent to the aggregator. Original reading of sensor is hidden and is not transferred to the aggregator as wireless medium is highly unreliable and insecure.

Table 3.1: Mapping Table

RANGE	PATTERN CODE
1-10	1
11-20	7
21-30	6
31-40	2
41-50	9

Data sent to the aggregator in encrypted form. The encryption process is same and known to every sensor in the network. But the key that is used in the encryption process is different for different category of sensor i.e. If aggregator belongs to the same zone, the zone key is used in the encryption process (zone key is common within zone but different for other zones within same cluster and similarly this key is different for same zone within different cluster). But if the aggregator not belongs to the same zone, then cluster key is used in the encryption process (cluster key is common within all members of same cluster but it is different for any member of other cluster. After receiving the encrypted data, cluster head or aggregator node decrypt the data with appropriate key (zone key or cluster key). Now the cluster head or aggregator sensor prepares the list of sensors that sent data to the base station. This selection is made on the basis of repetition factor, i.e. only one sensor is selected from the list of sensors that has the same reading range. Priority is given to those sensors that have transmitted the data to the base station for less number of times. The more is the repetition factor, the less data is transferred to the base station. When the selected sensors sent the data to the base station, sensor key is used to encrypt the data. This key is unique for all the sensors and only known to a particular sensor and a base station.

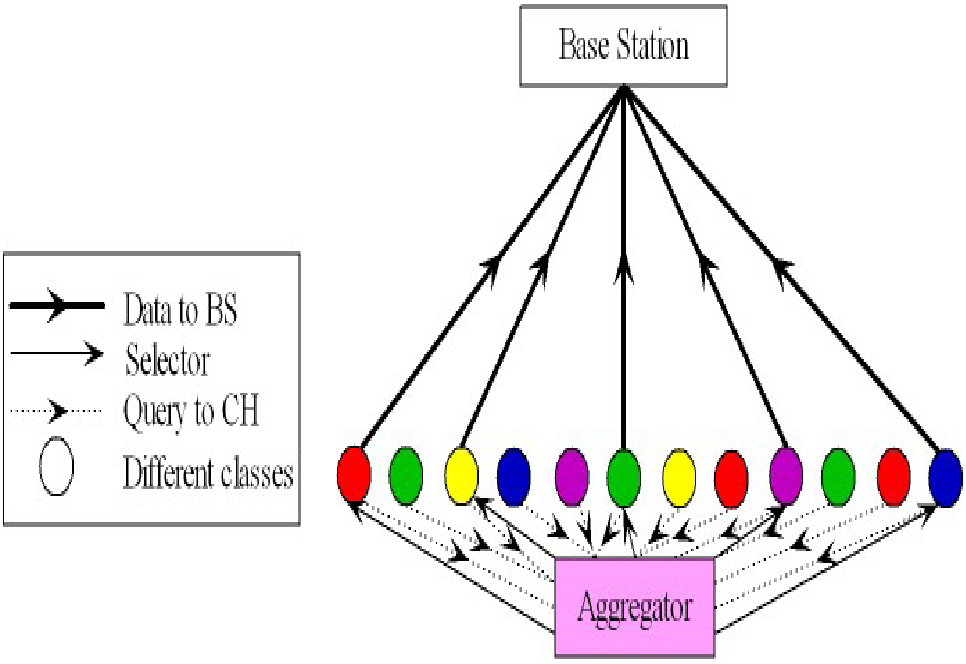

Figure 3.7: Data Transfer Scheme through Selector

The above figure (Figure 3.7) shows same class sensors with same color and different class sensors with different color. All the sensors sense the data and sent corresponding range value to the aggregator or cluster head (CH) for decision making in the form of query. This process is shown with dashed line. If multiple sensor nodes sent the same pattern code to the aggregator, then aggregator selects one target sensor from each repeating groups which is permitted to send the data to the base station. Aggregator informs target sensors that they are selected as a data transfer agent. This process is shown by solid line. These data transfer agents now sent the encrypted data to the base station.

3.1.3. Established Security Capabilities

To achieve security, the concept of dynamic (key is updated in every round) multi-key (Sensor Key, Zone Key and Cluster Key) is used in the presented scheme. This scheme is used in the system where every sensor is using more than one key. Each of this key is used for different purpose. For example sensor key is used when a sensor is elected as a data transfer agent. Similarly zone key is used when aggregator belongs to the same zone, otherwise cluster key is used. Every type of key is dynamic in nature, i.e. each type of key is updated after every round, randomly.

In key generation process, every sensor is provided with a virtual location. There are three types of virtual locations that are provided to every sensor in the network. These locations are virtual sensor location, virtual zone location and virtual cluster location. Virtual sensor location is used to generate the sensor key, virtual zone location is used to generate a zone key and similarly virtual cluster location is used to generate the cluster key. The process to generate a particular type of key (sensor, zone or cluster) is same except the virtual location.

If some sensor ‘A’ wants to transfer some information to a cluster head which is member of same zone within same cluster, then ‘A’ uses its zone key to transfer the information to the cluster head. Every sensor has an actual zone virtual location which is only known to all the members of the same zone and also to the base station. This location is used to generate a zone key.

If some sensor ‘A’ wants to transfer some information to a cluster head which is not a member of same zone within same cluster, but is a member of different zone within same cluster, then ‘A’ uses its cluster key to transfer the information to the cluster head. Every sensor has an actual cluster virtual location which is only known to all the members of the same cluster and also to the base station.

If some sensor ‘A’ is selected as a data transfer agent, then ‘A’ uses its sensor key to transfer the information to the base station. Every sensor has an actual sensor virtual location which is only known to sensor and Base station only. This location is used to generate a sensor key for a particular sensor.

A virtual origin is used in the network. Every sensor maps their virtual location over this virtual origin. The coordinates of this virtual origin is not (0, 0). For example if virtual location of any sensor is (5, 2) and the coordinates of virtual origin is (3, 2) then the virtual location of the sensor after mapping is (2, 0). Virtual origin is updated when all the sensor of the network is elected as a cluster head, i.e. when all the sensor of a network becomes the member of a group G. whenever a sensor is elected a cluster head, it becomes the member of a group G. Group ‘G’ has been set as null when all the alive members of a network becomes the member of group ‘G’.

3.2. KEY ASSIGNMENT COMPUTATION

The procedure to generate particular type of key (sensor, zone or cluster) is same except the virtual location that is used in the key generation process. The process is explained as follows:

3.2.1. Cluster Key Computation

Cluster key (KC_j) is calculated by applying a one way hash function on CVD1 (cluster virtual distance1) and CVD2 (cluster virtual distance2) as shown in Eq. (3.1).

$$Cluster\ Key\ (KC_j) = \int_H (CVD_1, CVD_2) \qquad \ldots \qquad (3.1)$$

Where CVD_1 is the distance between $CVLN_j$ (cluster virtual location of j^{th} cluster, calculated on virtual origin of the network and $CVL_{OLD}N$ (Cluster old virtual location, calculated on virtual origin (NVO) of network). CVD_2 is the distance between $CVLN_{j\_CURRENT}$ (cluster current virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) and $CVLN_{j\_OLD}$ (cluster old virtual location of j^{th} cluster, calculated on virtual origin NVO of the network).

Whenever every sensor of the cluster is elected as a cluster head, a new cluster virtual location CVL_{j_NEW} (cluster new virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) is provided to each cluster within network. Now $CVL_{j\_CURRENT}$ (cluster current virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) becomes the CVL_{j_OLD} (cluster old virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) as shown in Eq. (3.2) and CVL_{j_NEW} (cluster new virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) becomes the $CVL_{j\_CURRENT}$ (cluster current virtual location of j^{th} cluster, calculated on virtual origin NVO of the network) as shown in Eq. (3.3).

$$CVL_{j\_OLD} = CVL_{j\_CURRENT} \qquad \ldots \qquad (3.2)$$

$$CVL_{j\_CURRENT} = CVL_{j\_NEW} \qquad \ldots \qquad (3.3)$$

Whenever every sensor in the network is elected as a cluster head, a virtual origin NVO-$_{\text{NEW}}$ of the network is updated in the network, thus virtual location of each cluster on the network is also updated. So the key of every cluster is updated according to new virtual origin of the network.

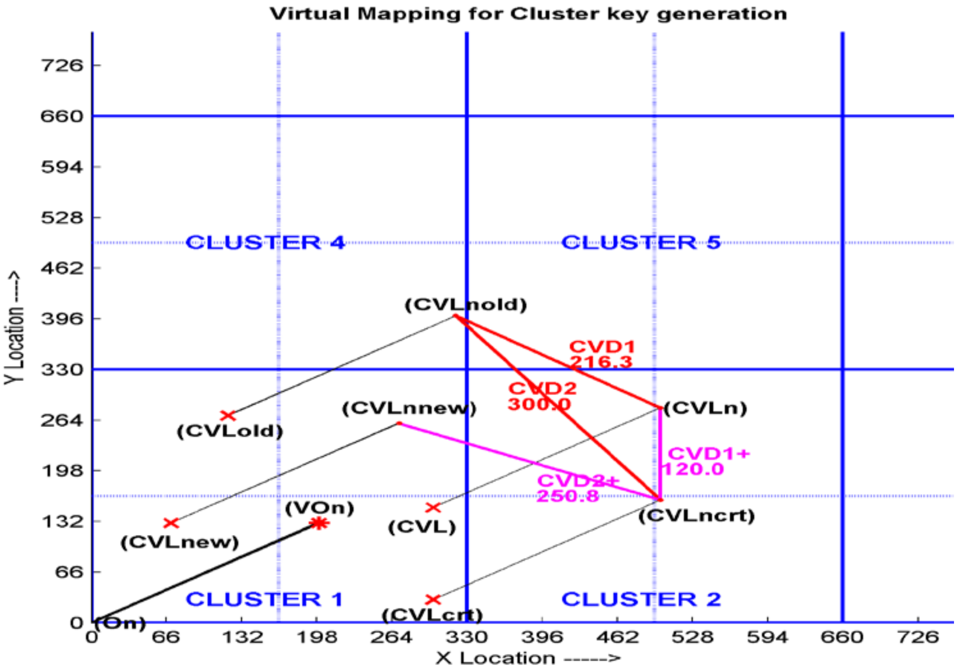

Figure 3.8: Virtual Cluster Location Updated to Generate a Cluster Key

3.2.2. Zone Key Computation

Zone Key $(ZK_{j,k})$ This key of k^{th} zone within j^{th} cluster is calculated by applying a one way hash function on ZVD_1 (zone virtual distance1) and ZVD_2 (zone virtual distance2).

$$Zone\ Key\ (ZK_{j,k}) = \int_H (ZVD_1, ZVD_2) \qquad \ldots \qquad (3.4)$$

Where ZVD_1 is the distance between $ZVLC_{j,k}$ (zone virtual location of k^{th} zone in j^{th} cluster, calculated on virtual origin of the cluster and $ZVL_{OLD}C$ (zone old virtual location calculated on virtual origin (CVO) of the cluster and ZVD_2 is the distance between $ZVLC_{j\_CURRENT}$ (zone current virtual location of k^{th} zone in j^{th} cluster, calculated on virtual origin CVO of the cluster) and $ZVLC_{j\_OLD}$ (zone old virtual location of k^{th} zone in j^{th} cluster, calculated on virtual origin CVO of the cluster).

Whenever every sensor of the cluster is elected as a cluster head, a new virtual zone location ZVL_{NEW} is provided to each zone within cluster, at the same time virtual origin of the cluster is also updated. Now $VZ_{CURRENT}L_{K,J}$ (Virtual current Zone location of k^{th} zone in j^{th} cluster, calculated on new virtual origin VO_j of j^{th} cluster) becomes the $VZ_{OLD}L_{K,J}$ (Old virtual zone location of k^{th} zone within jth cluster, calculated on new virtual origin VO_j of j^{th} cluster) and $VZ_{NEW}L_{K,J}$ (New virtual zone location of k^{th} zone within j^{th} cluster, calculated on new virtual origin VO_j of j^{th} cluster) becomes the $VZ_{CURRENT}L_{K,J}$ (Current virtual zone location of k^{th} zone within j^{th} cluster, calculated on new virtual origin VO_j of j^{th} cluster).

$$ZVL_{j,k\_OLD} = ZVL_{j,k\_CURRENT} \quad \ldots \quad (3.5)$$

$$ZVL_{j,k\_CURRENT} = ZVL_{j,k\_NEW} \quad \ldots \quad (3.6)$$

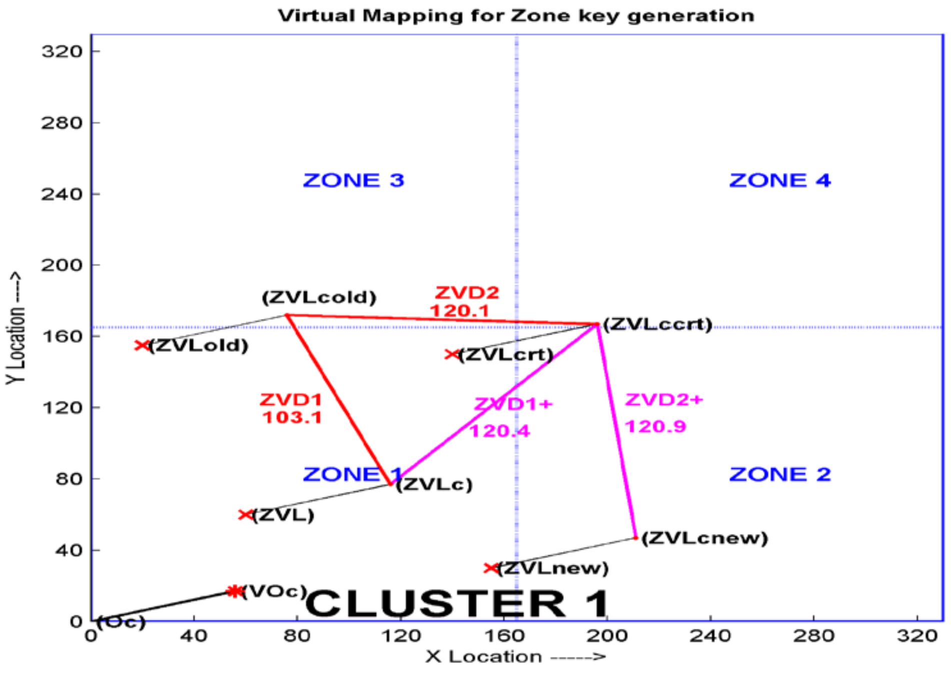

Figure 3.9: Virtual Zone Location Updated to Generate a Zone Key

3.2.3. Sensor Key Computation

Sensor key $(\mathrm{K}_{\mathrm{i}})$ is calculated by applying a one way hash function on VD1 and VD2.

$$Sensor\ Key\ (K_i) = \int_H (VD_1, VD_2) \qquad \ldots \qquad (3.7)$$

Where VD_1 is the distance between VLZ_i (Virtual location of sensor node i, calculated on virtual origin of the zone ZVO) and VLZ_{i_OLD} (old Virtual location of sensor node i calculated on virtual origin of the zone (ZVO) and VD_2 is the distance between $VLZ_{i\_CURRENT}$ (current virtual location of sensor node i, calculated on virtual origin of the zone (ZVO)) and VLZ_{i_OLD} (old virtual location of sensor node i calculated on virtual origin of the zone (ZVO)).

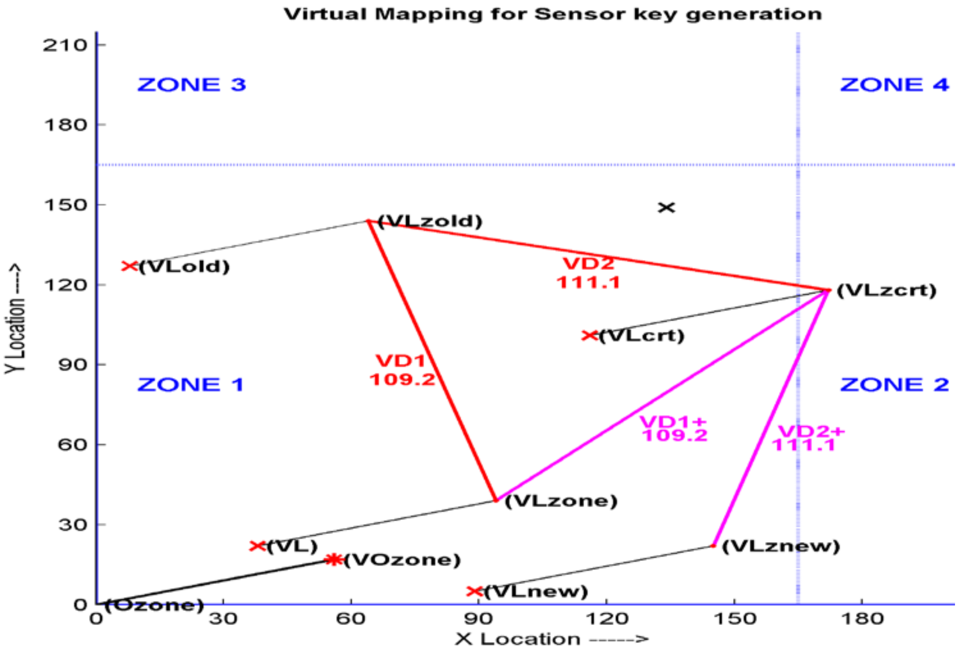

Figure 3.10: Virtual Sensor Location Updated to Generate a Sensor Key

Whenever every sensor within cluster is elected as a cluster head, a new virtual location VL_{NEW} is provided to each sensor within all zones in the cluster, at the same time virtual origin of all the zones is also updated. Now $VLZ_{CURRENT}$ (current virtual location of sensor, calculated on new virtual origin of the zone ZVO) becomes the VLZ_{OLD} (Old virtual location of the sensor calculated on new virtual origin of the zone ZVO) and similarly VLZ_{NEW} (New virtual location of the sensor calculated on new virtual origin of the zone ZVO) becomes the $VLZ_{CURRENT}$ (Current virtual location of the sensor, calculated on new virtual origin of the zone ZVO).

$$VL_{OLD} = VL_{CURRENT} \qquad \dots \qquad (3.8)$$

$$VL_{CURRENT} = VL_{NEW} \qquad \dots \qquad (3.9)$$

3.3.ENCRYPTION SCHEME

The scheme uses one way hash function to encrypt the data used in the network. This is a function that takes input 'I' and produce output 'O'. A special property of this function is that it is easy to produce 'O' with 'I' but it is completely impossible to produce 'I' with 'O'. This function is used in generation of any type of key. Generation of cluster key is illustrated with example

$Cluster\ Key\ (CK_j) = \int_H (CVD_1, CVD_2)$

$CVD_1 = 216$

$CVD_2 = 300$

$Cluster\ Key\ (CK_j) = \int_H (216{,}300)$

$\int_H$ (Hash function) are as follow:

a. Add CVD_1 and CVD_2.

   CVD=$CVD_1 + CVD_2$

   CVD=216+300=516.

b. Convert CVD into binary.
   BCVD=001000000100

c. Apply skip pairing scheme on BCVD. In skip pairing scheme, pairs are made on skip bits, i.e. first bit is paired with third bit and second bit is paired with forth bit and so on. The pair of ith bit is made with (i+2)th bit if (i mod 4) is one or two and the pair is (i, i+2), on the other hand if (i mod 4) is zero or three then pair of ith bit is made with (i-2)th bit and the pair is (i-2, i).

   SPBCVD=01, 00, 00, 00, 00, 10.

d. Apply Encoding Scheme. In this scheme, first output bit is 1 if both pairing bits are same otherwise 0 and second output bit is same as second pairing bit if both pairing bits are same otherwise it is the complement of second pairing input bit.

   ESPBCVD=001010101001

e. Convert ESPBCCV to decimal to generate the key.

   KEY= B2D (001010101001)

   Where B2D is a function which converts a binary number into equivalent decimal format.

So, Cluster key (CK_j) is 681. Same algorithm is used to generate sensor key and zone key.

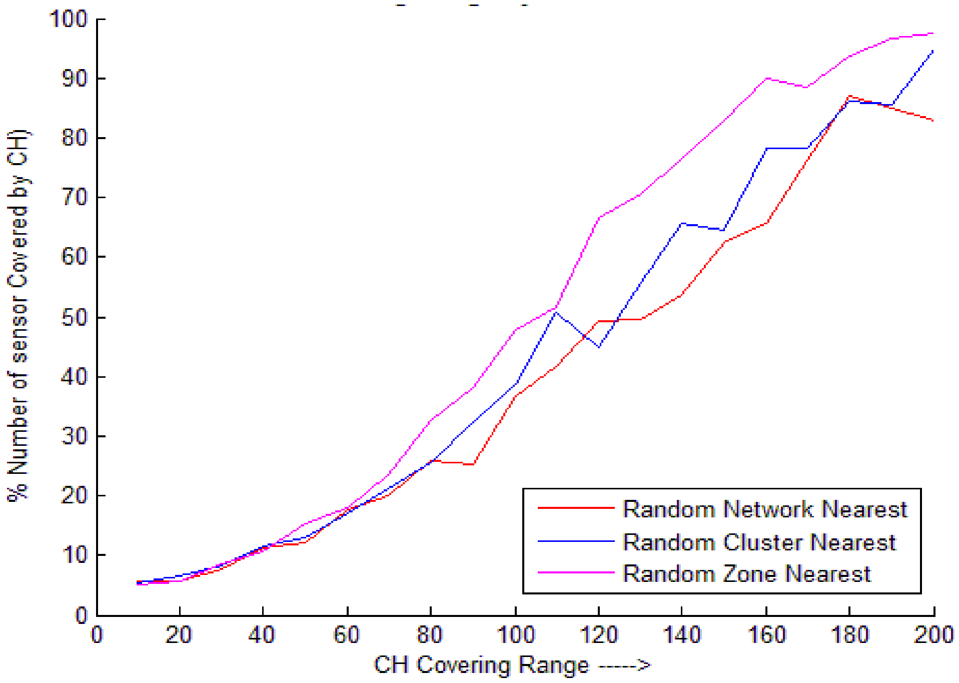

Figure 3.11: Sensor Coverage with Different CH Range

3.4. SIMULATION RESULTS

3.4.1. Sensor Coverage

Sensor coverage has been recognized as an important factor in clustering and data aggregation. To extend sensor coverage, one potential approach is to select a cluster head in such a way that each of which can cover unique set of sensors so that overlap area is minimized. The number of sensors that are deployed in a network is very large so a complete coverage is sometime not available. Simulation result has been calculated to check the coverage of sensors by cluster head in percentage under different scenarios. In one scenario cluster heads are selected randomly from the entire network (Random Network Nearest). In other scenario, the cluster head is selected randomly from each cluster (Random Cluster Nearest). In third scenario (presented approach), cluster head is selected randomly from the same zone within all clusters (Random Zone Nearest). If some sensor is under the coverage range of some cluster head, it is said to be covered. Covering range of cluster head is increased from ten meter to two hundred meter. Results in Figure 3.11 show that presented scheme is better in terms of coverage than the other two schemes.

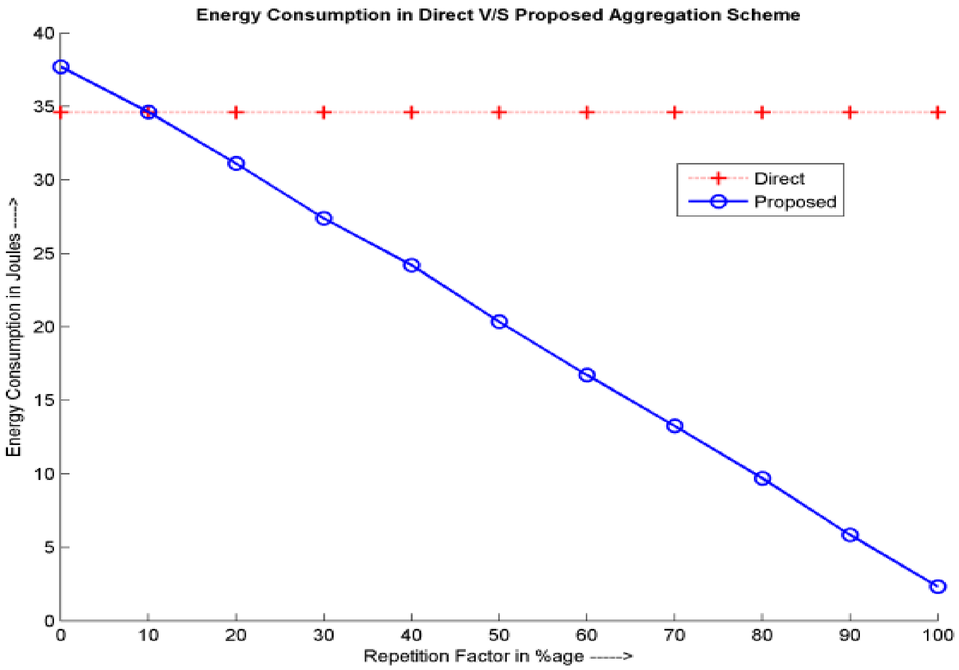

Figure 3.12: Energy Comparison in Direct v/s Presented Data Aggregation Scheme

3.4.2. Energy Consumption

This section presents the evaluation of presented scheme. To evaluate the performance of presented scheme, the energy consumption of data aggregation is calculated and compare with direct data transfer scheme. Simulation results proves that when the repetition factor is minimum, i.e. the reading of every sensor in the network is unique then direct data transfer scheme is energy efficient as compare to presented scheme. This is because memory is wasted in compare the reading by aggregator or cluster head. On the other hand, when repetition factor increases in the network, presented scheme is energy efficient as compared to direct data transfer scheme. The energy consumption is shown in Figure 3.12.

3.5. SUMMARY

This chapter introduced an energy-efficient secure data aggregation scheme. In contrast to direct transmission scheme, presented scheme avoids the transmission of redundant data to base station with the help of cluster head nodes. The protocol also maintains the principle of confidentiality between a sensor node and the base station because in this protocol, the actual sensor reading is hidden from the cluster head node. This scheme also provides a security framework between sensor nodes and aggregator and also in between sensor node and the base station. Results prove that when the redundancy factor increases in the network, energy consumption is decreases automatically to improving the performance of the system.

In the next chapter, a model for key management scheme has been presented that overcomes the problem of key distribution in each round to each and every node in the network.