Efficient Data Aggregation Scheme in Secure Tree based Wireless Sensor Networks (EDAS)

Khalid Khan
Institute of Computing, Kohat University of Science & Technology, 26000 Kohat, KPK, Pakistan
khalid.iiu.ibrar@gmail.com

Abstract

Wireless sensor networks (WSNs) have emerged as an important area of research from the last two decades. In WSNs, sensors have limited memory, computational capability and battery power. One of the solutions to these issues is to reduce the data transmission and today, various data aggregation protocols have been proposed to reduce network traffic. But they created security issues like authentication, data integrity and freshness. Hop-by-hop secure data aggregation protocols were proposed to provide security along with data aggregation function. But data on the aggregators still need to be secured because aggregators are exposed to node compromises. End-to-end secure data aggregation protocols are proposed. They provide end-to-end security and privacy of data. One of these protocols is Secure End-to-End Data Aggregation (SEEDA), which ensured end-to-end privacy of data. However, the protocol has higher communication cost and computational overheads because it sends extra bits/data regarding non-responding nodes and performs unnecessary computations for non-responding nodes. I proposed new protocol, EDAS which reduces 12% to 25% communication and computational overheads as compared to SEEDA and it is also efficient in energy consumption.

Keywords: Wireless Sensor Networks, WSNs, Privacy, Secure Data Aggregation, Homomorphic Encryption

Article Number: IOARP-JCN04-002
IOARP Journal of Communication and Networks
Volume 4, Issue 1, June 2024
Appeared on Digital Library: 15 July 2024

1. Introduction

Today, WSN is the most popular wireless technology, and it has gained interest of researchers from the last twenty years. A WSN consists of small size sensor nodes and usually they are distributed randomly. Sensors are used to detect physical changes or environmental conditions, such as sound, temperature, air pressure and intrusion detection. The detected event or sensed data is passed through other sensor nodes and finally sent to the destination or sink node for taking all necessary actions. Some of the sensor networks also monitor sensor node activity. WSN was initially deployed in military applications such as surveillance of the battlefield. Nowadays WSNs are used in many industrial and consumer applications such as process control in industries, monitoring of machines etc. AWSN is composed of sensor nodes which range in number from a few sensor nodes to several hundreds or even thousands of sensor nodes, where each node is connected to one or more than one sensor nodes. These sensors are resource constraints such as limited low power, limited memory and computational resources [1,2]. Due to these limitations, WSNs are highly exposed to many security threats. The invention of micro electromechanical systems (MEMS) technology, low power and low-cost digital signal processors (DSPs) and radio frequency circuits have much contribution in the WSN technology [3,4]. The most important challenge in WSNs is to increase the lifetime of sensor node's battery. As I know that generally sensor nodes are deployed in harsh environment, and it is not possible to change their batteries such as in the case of battlefield.

Now it is very important to keep in mind that how to decrease the computational operations of nodes during the designing process of various protocols. It is not only enough to reduce computational operations but at the same time, it is also important to make these communication protocols energy efficient as well. Among the various protocols of WSNs, the data transmission protocols are needed to consider properly in order making them energy efficient because 70% of the total energy is consumed in data transmission [5]. In order to minimize the energy consumptions of the sensor nodes, packets sent across the network must be reduced. So, these applications may employ an in-network mechanism called 'Data Aggregation'. Data Aggregation is the collection and combination of data from various sources and reduction of data size by using functions such as suppression (eliminating duplicates), sum, finding minimum or maximum value, averages, variance and standard deviation [6]. Aggregation protocols are used to minimize the amount of data communication in the network which conserves battery power. As data are detected and sensed by individual sensor, they need to be collected and processed. One of these approaches for sensor network is to send sensor's data to certain special nodes, the Aggregators. Each aggregator, aggregate the data before sending to other nodes. The aggregating nodes may either be special and more powerful nodes, or they may be regular sensor nodes. Data aggregation techniques can greatly contribute to conserve the limited energy resources of sensor network by removing data redundancy and minimizing the number of bits or data transmissions.

Sensors can also sense sensitive data; therefore, data needs high level security for protection purposes. Therefore, data is sent to base station in encrypted form. At the aggregators, the data is decrypted for aggregation, and then re-encrypted and sent towards the base station. Again, the data decryption at the aggregators is risk full. End to end secure data aggregation is used which ensures that data is aggregated without performing decryption at the aggregators. This is only possible with the use of additively homomorphic encryption technique that allows aggregation over encrypted data. Additively homomorphic encryption technique is introduced in [7,8] and it is used to make the aggregate of encrypted data without being decrypted. Such type of homomorphic encryption scheme is useful in problem scenarios where there is no decryption key, and someone want to perform arithmetic operations on a set of cipher texts.

The main emphases in data aggregation protocols for WSNs is on ensuring security, reducing computational and communication costs. The ultimate result is increased lifetime of the wireless sensor networks. The existing protocol SEEDA [9] ensures end-to-end privacy of data but this protocol creates sufficient communication and computational overheads by computing and sending of cipher text for non-responding nodes. I emphases on these two issues and assume the network and security model to be same as used in SEEDA.

The contribution of the proposed scheme (EDAS) is to transmit less numbers of bits/data than existing scheme SEEDA, because data transmission consumes 70% energy in wireless sensor networks. Also, my goal is to decrease the computational cost created on the aggregators. By decreasing computational and communication cost in wireless sensor network, I can prolong the lifetime of WSN.

The rest of the paper is organized as: in section 2 a detailed related work is presented. In section 3 and 4, the problem statement and proposed solution are presented respectively. Implementation scenarios are presented in section 5. Results and comparison parameters are presented in sections 6 and section 7 concluded the paper.

2. Related Work

To reduce the data transmission through the network a Pattern Based Energy Efficient Protocol (ESPDA) [10] is proposed by H. Cam and S. Ozdemir. This protocol reduces data transmission through the network by sending patterns of data instead of sending whole data to the base station. The pattern codes reflect characteristics of original data sensed by sensors. ESPDA is a cluster-based approach where multiple sensors are attached to it. The sensor nodes sense data from the surrounding environment and then make patterns of the sensed data. The patterns are generated in such a way that they represent the sensed data in all respect. Patterns are then sent to cluster head. Cluster head compare the patterns received from various sensors and send only those patterns which are distinct. The redundant patterns are removed here. This protocol also has the capability to sleep those sensors which sense redundant data from the environment. It results in energy conservation of the sensors and increase network lifetime. For security purpose, ESPDA refreshes the mapping interval of patterns code periodically. ESPDA has some limitations as well along with benefits. This protocol maintains a lookup table for pattern generation in each sensor. For storage constrained sensors, this lookup table is wastage of storage. Second drawback is the recalculation of lookup table time and again after specific interval. It creates computational overheads. The sensor always searches for a critical value in the look up table each time a new data is sensed which again results in computational overheads. The third limitation is that ESPDA do not allow intermediate aggregation. It only allows immediate aggregation.

SRDA, Secure Reference based Data Aggregation protocol [11] is proposed by H. Sanli and S. Ozdemir. In this protocol the base station sends reference value to sensor network. Sensors make the deferential data of the original data and reference value. So differential data is then sent to the base station instead of raw data due to which, number of bits transmitted in the network are reduced. For security purpose, SRDA uses an algorithm with consideration of security margin as adjustable parameter. In the first step sensors send raw data to the cluster head. The cluster head compares the raw data with reference value. Now the differential value is sent to higher level. The reference value is refreshed for security reason after session expires. The new reference value is calculated as average of the previous sensor readings. In the conventional data aggregation protocol, raw data was sent by sensors to cluster head which is energy consuming process. Reducing data transmission in wireless sensor networks is essential. To explain SRDA, let us take an example. Let the temperature measured by sensor is 105 F, and reference value is 100. Then differential value will be 5 F. The efficiency of the SRDA will be greater if reference value is nearer to the sensor readings. The drawback of SRDA is same as that of the ESPDA. It also does not allow intermediate node to perform data aggregation. It only allows immediate node to perform aggregation function which significantly reduces the importance of data aggregation.

In paper [12], a Secure Hop-by-Hop Data Aggregation Protocol (SDAP) is proposed by Y. Yang and X. Wang. The authors of the SDAP realized that most of the trust is placed on higher level nodes as compared to low level nodes, which is not a wise decision. So, if the compromised node is at higher level, then there will be more chance of false results at base station. This is because higher level nodes carry larger aggregated data then lower level nodes. There is need of method which could reduce trust on the higher level nodes. SDAP has reduced the trust on higher level nodes using divide-and-conquer principal. It uses probabilistic approach and dynamically partitions the topology tree into multiple sub-trees of equal size. Since in this approach, fewer sensors are attached to each higher level node so the security risk of larger data being compromised is reduced. SDAP provides data integrity, source authentication and confidentiality. Along

with these benefits, this protocol has also achieved applicability on multiple aggregation functions. The limitations of this protocol include greater amount of energy utilization and transmission overheads. In SDAP data is decrypted at aggregators for aggregation function. At this stage is data is open to intruders and can be captured easily. Moreover, extensive encryption and decryption at the aggregators results in computational overheads.

The problem of Hop-by-Hop encryption and decryption is solved by Concealed Data Aggregation (CDA) [13] proposed by D. Westhoff. This protocol is based on symmetric cryptographic key which mean same secret key is used throughout the network. Data sensed by sensor is encrypted using this secret key. Encrypted data is sent to aggregator. Aggregators do not decrypt the encrypted data received from sensors. Due to Privacy Homomorphism (PH), aggregation function is applied on encrypted data. This process leads to get rid of costly decryption and encryption functions at each aggregator. CDA achieved end-to-end privacy of data. The aggregators are required not to store the secret key because they do not decrypt the data. In this manner the memories of aggregators are conserved. The limitation of this approach is the sharing of single secret key throughout the network. If a single node is compromised, then intruder can easily capture the data of all the sensors. This technique is also vulnerable to reply attack.

A secure data aggregation protocol called CDAP [14] is proposed by S. Ozdemir which uses asymmetric based privacy homomorphic cryptography for achieving end-to-end to confidentiality along with data aggregation. The author of this paper realized that privacy homomorphism based on asymmetric cryptography carry high computational overheads. For resource constrained sensors cryptography is not affordable. To solve this problem, CDAP employee resource rich nodes called AGGNODEs (aggregator nodes). The function of these AGGNODEs is to carryout privacy homomorphism and aggregation. Each of the neighbours encrypts its data by using encryption: RC5 and sends this encrypted data to the AGGNODE. Now the AGGNODE decrypt all the received data which it gets from its neighbours, aggregate them and by using PH the aggregated is encrypted. This encrypted data is then sent to base station. The encrypted data with PH encryption technique can only be decrypted by the private key of base station only. During forwarding process, due to PH property, the intermediate AGGNODE can aggregate the encrypted data. Privacy Homomorphism creates extra computational overheads then hop-by-hop schemes. However, data transmission efficiency and aggregation ability of this scheme increase with the increase in number of AGGNODEs.

Hierarchical Concealed Data Aggregation protocol (HCDA) [15] proposed by S. Ozdemir and Y. Xiao. This protocol aggregates the data hierarchically using elliptic curve cryptography for security purpose. Analysis shows that this scheme is resistive to node compromise attack. This protocol achieved concealed data aggregation and aggregate data of multiple sensor node groups where each group uses different public key to encrypt data. In this scheme the nodes deployed in groups where sensors are divided into multiple groups before deployment. Each of the groups is deployed over certain location for covering the whole area of interest. Each group is assigned with a separate public key to encrypt data. Based on this public key, the base station would easily determine that to which group this data belongs and also easily classify the data. Elliptic curve cryptography is better than public key cryptography from communicational, key size and memory point of view.

In [8] a new scheme Efficient and Provably Secure Aggregation of Encrypted Data based on homomorphic encryption is proposed. This protocol allows an aggregator to perform the aggregation function and aggregate the encrypted data which it receives from its sensors with no need for decryption or to make the plain text messages at the aggregators. Here in additive homomorphic function the aggregators do not decrypt or recover the original plain text message received from its children. Since the data is in cipher text form so even if an aggregator is compromised still data is safe and cannot be read. Due to this reason, much stronger privacy is achieved than a simple aggregation scheme using technique like hop-by-hop encryption. From bandwidth point of this scheme is less efficient than the hop-by-hop scheme. Beside this it provides a stronger level of end-to-end privacy comparable to that provided by end-to-end encryption with no aggregation. In this scheme the communication load is distributed evenly among all other nodes. This results in longer network lifetime. The limitation in this scheme is the generation of significant

overhead if some of the sensors do not respond. Number of bits (data) is increased due to sending of additional information related to non-responding nodes.

To decrease number of bits transmitted as well as to maintain the End-to-End privacy of data in the network, a Secure End-to-End Data Aggregation protocol (SEEDA) is proposed [9]. The SEEDA protocol is applied to tree based wireless sensor networks. The scheme ensures end-to-end data privacy. Less number of bits or data is transmitted from sensors to base station as compared to end-to-end aggregation protocol [8]. SEEDA protocol sends smaller amount of data which the best feature of hop-by-hop. It also adopts the best feature of end-to-end protocols which is end-to-end privacy of data. This scheme in this paper claims that the average data which is transmitted per node is reduced from 30% to 50% as compared to the scheme in [8]. The working of this protocol is explained as below.

- Responding sensor nodes sense data and encrypt using function $\boldsymbol{c_i = x_i + k_i\ (modM)}$ and send this encrypted data to the aggregators.
- Aggregator nodes add the encrypted data received from all responding nodes using Additive Homomorphic encryption i-e $\boldsymbol{C_{agi} = C_{agi} + c_i}$
- For non-responding nodes, the aggregator node encrypts zero data using function $\boldsymbol{c_j = 0 + k_j(modM)}$. This cipher text of jth non-responding node is also added with $\boldsymbol{C_{agi}}$ i-e $\boldsymbol{C_{agi} = C_{agi} + c_j}$. Now the ith Aggregator node will construct the message as m_i = $\boldsymbol{C_{agi}||q_i}$ where $\boldsymbol{q_i}$is count number of non-responding nodes.
- This message is forwarded to the aggregator at higher level of tree where the same homomorphic encryption is performed. This process is continued until message reach to the base station.

SEEDA protocol encrypts zero data at aggregators for non-responding nodes, $\boldsymbol{c_j = 0 + k_j(modM)}$. Since, this protocol computes the average of the data that requires data of responding nodes only, at the sink node. So, the need of caring out of the function and adding it with message is no more required. Due to encryption of zero data of non-responding nodes, the number of bits transmitted in the network is increased. The computation overhead is also increased on the aggregators due to computing the function,$\boldsymbol{c_j = 0 + k_j(modM)}$ for non-responding nodes. Extra energy is consumed, and the life of the network is decreased.

Concealed data aggregation (CDA) [13] has much more importance in wireless sensor networks. Because the algorithms based on Privacy Homomorphism (PH) are basis for CDA. But the limitation here is that it sends whole encrypted data to the Base Station. This process is energy consuming from communication point of view. An Energy Efficient Recoverable Concealed Data (EERCDA) is proposed by Josna J [16]. This technique reduces the energy burden on the sensors imposed by the recovery based concealed data aggregation by transferring signature and cipher text both. The EERCDA transfer differential data from nodes to the Base Station instead of sending whole encrypted data. In this way the cluster head also has to process small amounts of bits/data. EERCDA gives energy relief to sensor nodes as well as the cluster head. Beside energy befit, it also preserves the security requirement such as source authentication, integrity and confidentiality.

Due to excessive encryption and decryption at the aggregators, the Hop-by-Hop aggregation protocols incur larger delays to data. This operation is not suitable for time crucial and delay sensitive military applications. To overcome delay problem, a protocol PEPPDA: Power Efficient Privacy Preserving Data is proposed by Joyce J [17]. This protocol reduces delay by allowing data aggregation function on encrypted data. It means data is not decrypted on the aggregators. Along with reduced delay PEPPDA also claims confidentiality, data freshness and accuracy. The protocol also claims low communication and computational costs.

A novel protocol called Dynamic Data Aggregation for Energy Optimization in Multi-Hop Wireless Sensor Networks [18] is proposed by Abhilash L N and Devaansh Goenka. This protocol optimizes the energy consumption of sensors in a multi-hop sensor network by using a new data aggregation technique. This technique discarded unnecessary packets to reduce data transmission and energy consumptions. This process is of two steps: in first step Exponential Weighted Moving Average (EWMA) data aggregation

technique is used to compare the current data value with all the previous values before deciding whether to forward or drop the packet. The second step optimizes the network even further by considering readings from neighbouring sensors into the equation. This protocol claims 20% reduction in data transmission. It also reduces bandwidth requirements of the networks.

3. Problem Statement

The problem statement consists of problem scenario 1 and 2, discussed in the following in sub-sections.

3.1. Problem Scenario 1 (Communication Cost)

For non-responding nodes, the aggregator node encrypts zero data using function $\boldsymbol{c_j = 0 + k_j (mod M)}$. Now this cipher text of j[th] non-responding node is also added with $\boldsymbol{C_{agi}}$ i-e $\boldsymbol{C_{agi} = C_{agi} + c_j}$. Now the i[th] Aggregator node will construct the message as m_i = $\boldsymbol{C_{agi}||q_i}$ where $\boldsymbol{q_i}$is count number of non-responding nodes. So here it is clear from the above scenario 1 that adding $\boldsymbol{c_j}$for non-responding with the message increases the message size. This will increase number of bits (communication cost) transmitted in the network. For example, if cipher text size for one non-responding node calculated by the aggregator is 16 bit, then the total size of cipher text for all non-responding nodes at a single aggregator will be calculated as Total size = number of non-responding nodes x 16 bits. The number of bits will increase as it goes upward through the tree.

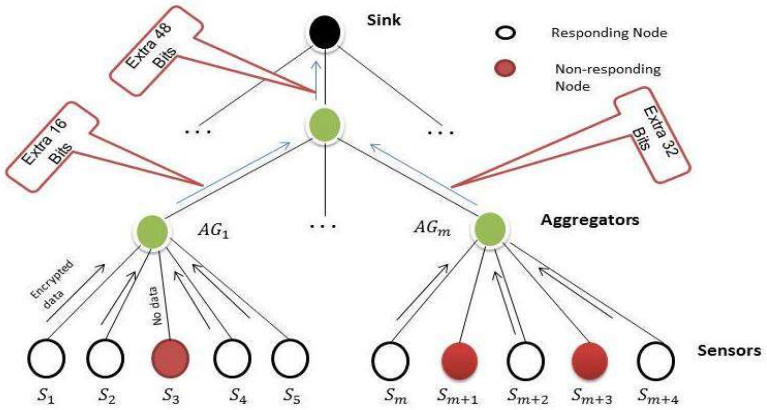

Figure 1: Problem scenario 1 (Communication Cost)

3.2. Problem Scenario 2 (Computational Cost):

At the aggregators the function is computed for all the non-responding nodes as $\boldsymbol{c_j = 0 + k_j (mod M)}$ which also increases the computation cost. For example, in the above scenario 2, if computational energy cost for single non-responding node is 0.2 Micro-Joules. Then the total computational cost for all non-responding nodes at single aggregator will be as Total Energy Cost = number of non-responding nodes x 0.2 Micro-Joules

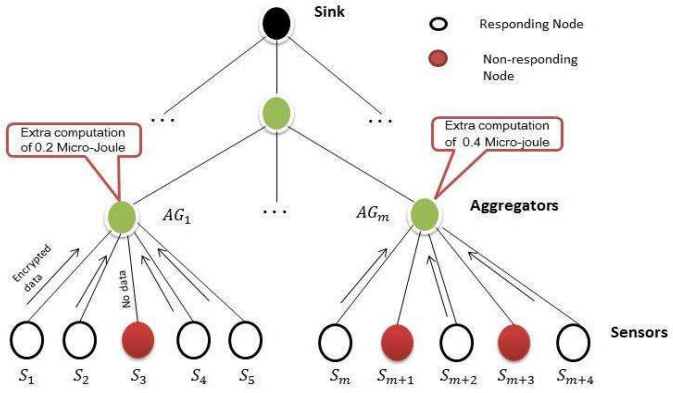

Figure 2: Problem scenario 2 (Computational Cost)

4. Proposed Solution

This section is related with detailed explanation of assumptions made in the research, network model, algorithms used in the scheme and flow chart of the protocol.

4.1. Assumptions in EDAS

In EDAS I have made the following assumptions.

- The network model is same as that of the SEEDA and tree based fixed network is used.
- All nodes are potential aggregating nodes except leaf nodes which are sensors.
- The security model is same as that of the SEEDA.
- Key management and key distribution are also assumed to be same as SEEDA.

4.2. Network Model

The network model in my scheme is same as that of the SEEDA. A multilevel heterogeneous network tree is used. This tree consists of sensors, aggregators (AGGs) and a base station or sink node as shown in figure 3. Sensors are at the leaves i-e level 'h' of the tree. The aggregators are located in all levels of tree except level 'h' and level zero. The sensors have limited energy resources while that of aggregators have greater energy resources. Sensors sense data from the surrounding environment. The aggregators are responsible for forwarding queries and receive data from the sensors. They are also responsible for aggregating data received from various sensors. In the figure 3 the nodes S_0, S_1, S_2……S_{2m} are sensor nodes and the nodes SGl to SGM are aggregating nodes.

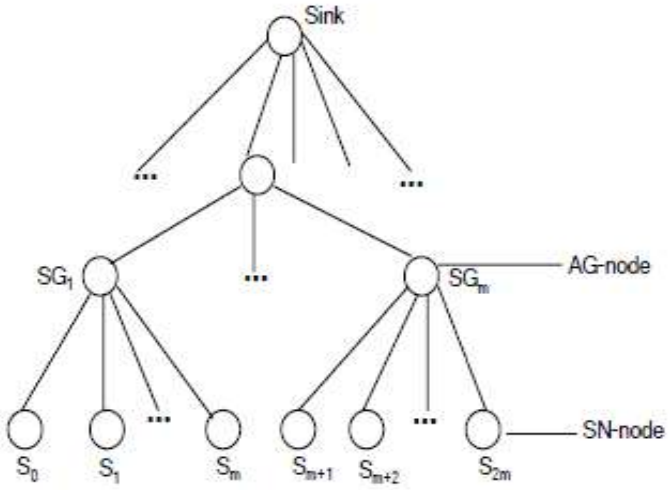

Figure 3: Network Model

4.3. Methodology

Each sensor node *Si* will sense data and encrypt using function$ci = xi + ki\,(modM)$. This encrypted data will be sent to the corresponding aggregator node, AGG. The aggregator node will collect encrypted data from all the concerned sensors and will make the aggregate without decrypting data by using additive homomorphic encryption technique. This will ensure security and end-to-end privacy of data. For non-responding nodes the aggregator will append the ID of that node and will make the status of that node to zero which mean that this node will no longer be considered in later rounds. Here SEEDA encrypt zero data for non-responding nodes and add with the message which is more energy consuming process in terms of communication and computation. Now the aggregator will make message by appending IDs of non-responding nodes with aggregated data. The message will be sent to the aggregator at next higher level of the tree. Aggregator at this level collect messages form concerned aggregators and make aggregate and forwards the message to higher level. This process will be continued until message reached to the base station (BS) or Sink.

4.4. EDAS Algorithm

Step 1: Sensor *Si* sense data, encrypt using function $ci = xi + ki\ (mod\ M)$

Step 2: Sensor send data to the concerned *AGG*

Step 3: *AGG* checks: if sensor node is responding

a). Yes: perform aggregation function i-e $C_{agi} = C_{agi} + c_i$

b). No: Extract ID and make status of node to zero

Step 4: Make message M: aggregated data appending with IDs of non-responding nodes

i-e$M = C_{agi} || ID_i$

Step 5: Send message to AGG on next level

Step 6: *AGG* make aggregate of the messages received from other *AGGs*

Step 7: Repeat steps 5 and 6 until message reach BS

Step 8: END

Explanation: In step 1 of the EDAS algorithm, sensor nodes will sense or detect data from the environment. This detected data will be the encrypted using the function mentioned in the step 1.

In step 2, sensors will send the encrypted data to the concerned aggregator *AGG*.

In step 3 of the algorithm, the aggregators will check that all sensors are responding or not. For responding sensor, the aggregator will make aggregate of their sent data using Additive Homomorphic Encryption. For non-responding sensors, aggregator will find the ID of that sensor and will make the status of that node to zero. Here zero status means that this node will be marked as permanently non-responding and no processing for that node in future.

In step 4, aggregator will make message by appending aggregated with the IDs of non-responding sensors.

Step 5 states that the aggregators at level (h-1) of the tree will send the message to aggregators at level (h-2). Where '*h*' is the height of tree.

In step 6, aggregators at level (h-2) will make aggregate of messages received from aggregators of underlying level.

Step 7 will continue repetition of step 5 and 6 till the message reach the base station.
Step 8 is the end of EDAS algorithm.
The above discussion is also clearer in EDAS flow chart shown in figure 4.

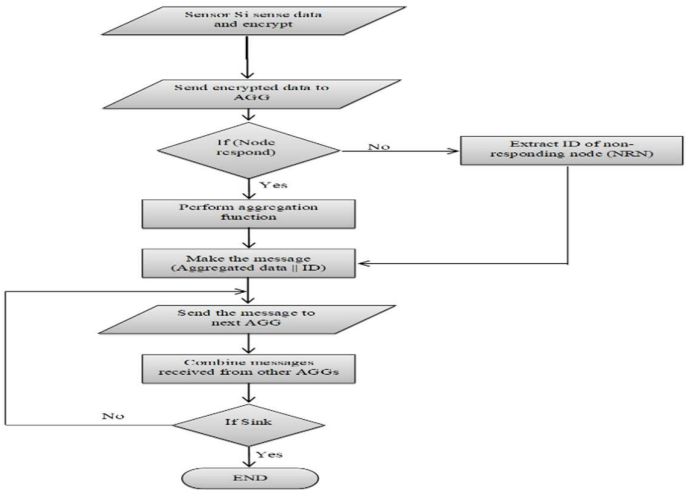

Figure 4: EDAS flow chart

5. Implementation

This section gives a detailed overview of implementation. At the start, implementation of proposed model, (EDAS) is explained. Next to it, the implementation of existing model, SEEDA) is explained.

5.1. Implementation Environment

EDAS and SEEDA simulators are developed by using Java JDK 1.7 and Net Beans IDE is used for Java JDK 1.7. Java is a programming language. It contains fundamental technology that enables high-technology programs containing utilities, games, mobile applications and business applications.

5.2. Implementation Scenario

Efficient Data Aggregation Scheme in Secure Tree Based Wireless Sensor Network (EDAS) Simulators shown in figure 5. This is just a control form where one can select or call either EDAS simulator or SEEDA simulator. By clicking the button labelled as EDAS, the corresponding simulator is called and displayed on the screen. By clicking the button labelled as SEEDA, the corresponding SEEDA simulator is displayed on the screen. And finally clicking the View Graph button, the Graph Form is displayed. This main EDAS Simulator form can be closed by clicking Exit from the File menu.

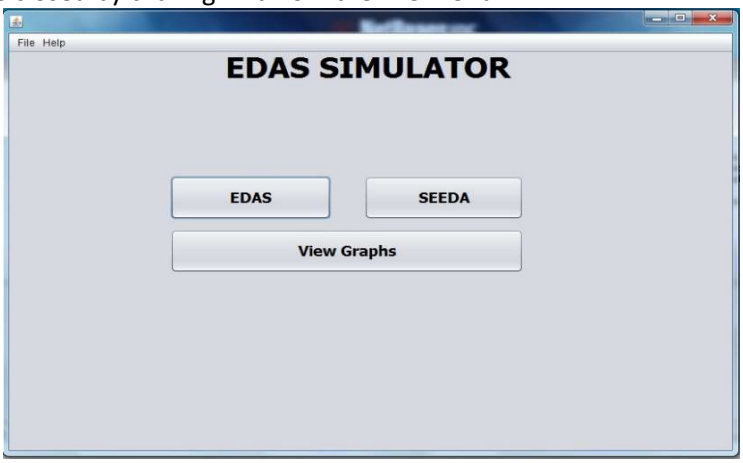

Figure 5: Main Form/EDAS Simulator

5.2.1. EDAS Simulator

In figure 6 EDAS Simulator is shown. In the panel of the simulator, a tree based network shown. At Level 0 of the tree is a node called Base Station. Aggregators are at Level 1 and Level 2. The leaf nodes are the Sensors. There are two radio buttons at bottom. The one labelled as 10% Non-Responding Nodes, when selected then 90% nodes in the network will be responding nodes and 10% will be non-responding. The radio button labelled as 30% Non-Responding Nodes, when selected then 70% of the nodes will be responding and 30% will be non-responding. There is a control button labelled as Start. The function of this button is to start the simulation.

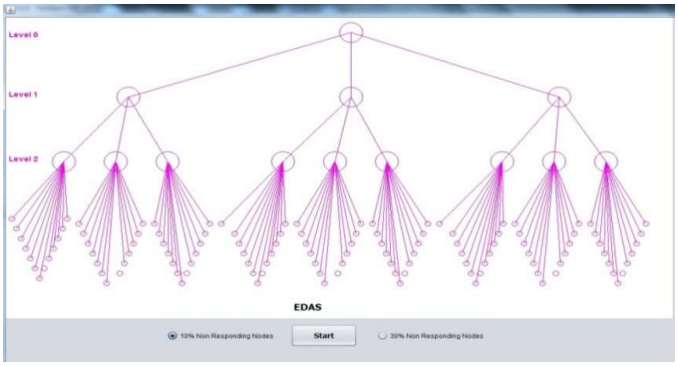

Figure 6: EDAS Simulator

5.2.2. SEEDA Simulator

In figure 7 SEEDA Simulator is shown. In the panel of the simulator, a tree based network shown. At Level 0 of the tree is a node called Base Station. Aggregators are at Level 1 and Level 2. The leaf nodes are the Sensors. There are two radio buttons at bottom. The one labelled as 10% Non-Responding Nodes, when selected then 90% nodes in the network will be responding nodes and 10% will be non-responding. The radio button labelled as 30% Non-Responding Nodes, when selected then 70% of the nodes will be responding and 30% will be non-responding. There is a control button labelled as Start. The function of this button is to start the simulation.

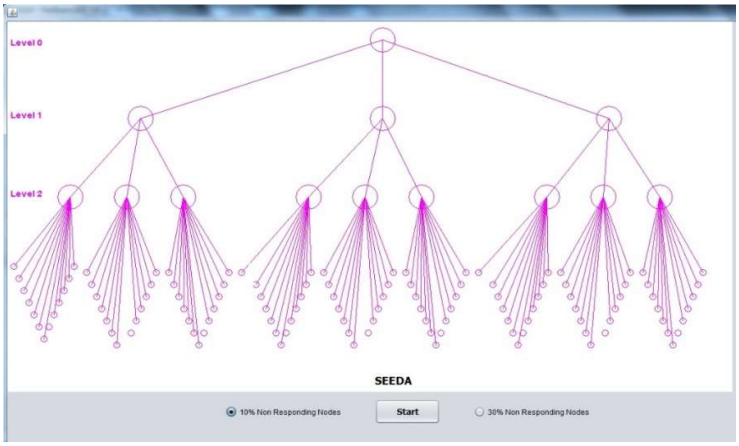

Figure 7: SEEDA Simulator

5.3. EDAS Parameters

The proposed EDAS scheme has 108 Sensors and 12 Aggregators 'AGG' and a single Base Station. Network used is heterogeneous i-e having different types of nodes.
The details of various parameters used in EDAS scheme are given in the table 1.

Table 1: EDAS Parameters

S.NO	Parameter	Value
1	Data packet size	232 bits
2	Header Size	56 bits
3	Data rate	250 k-bits/s
4	Orientation	2D
5	Propagation Environment	Free Space
6	Sensor Nodes	108
7	Aggregators	12
8	Base Station	1
9	Transmission range of sensor node	Up to 300 m
10	Radio frequency	915 MHz
11	Transmit mode Energy consumption	0.5 Milli-Joule
12	Receive mode Energy consumption	0.2 Milli-Joule
13	Computational Energy consumption	0.4 Micro-Joule

6. Results and Comparison Parameters

This section is related with simulation methodology such as assumptions of the simulation, mechanism of the simulation and simulation results. The results are used to show the efficiency and reliability of the existing technique SEEDA and that of the proposed scheme EDAS in terms of communication, computation, energy, latency and accuracy.

For the simulation of both schemes, I used Net Beans as IDE. For building custom simulator, java is used as programming language and JDK 1.7 is used as compiler. Results of the existing solution are studied, observed deeply and thoroughly. Then the results of the proposed solution are compared with the results of the existing technique. The results of the proposed scheme are better as compared to the existing technique.

I have used the following performance metrics in my scheme:

1. Number of bits/data sent to Base Station
2. Computational energy
3. Energy consumed by the sensors nodes in the network
4. Network lifetime
5. Latency/Data Delay
6. Data Accuracy

The first four graphs are plotted when 10% of sensors are non-responding. The last four graphs are plotted when 30% of the sensors are non-responding. Finally, the graph of network latency and data accuracy are plotted for various percentages of non-responding nodes.

6.1. Number of Bits Sent to the Base Station (10% Non-Responding Nodes)

The graph shown in figure 8 is used to represent number of bits sent to the Base Station when 10% of the sensors do not respond and 90% sensors respond. In graph the horizontal axes represent number of rounds while the vertical axes represent number of bits/data sent to the base station. This graph compares EDAS with the SEEDA scheme. It is clear from the plot that in SEEDA scheme, the base station receives more than 150 extra bits than EDAS scheme in each round. Round means data sensed by all the responding sensors and sent to the base station. These extra bits are produced in SEEDA, because the aggregators compute cipher text c_j for all non-responding nodes and add this data with aggregated data. The red line in the plot is for SEEDA scheme while the blue line is for EDAS scheme.

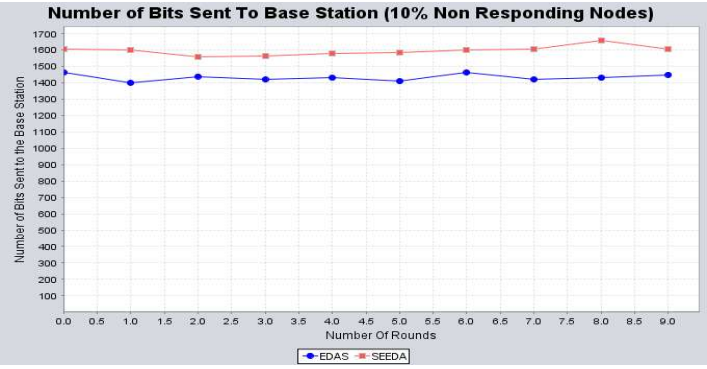

Figure 8: Number of Bits sent to Base Station (10% non-responding nodes)

6.2. Computational Energy Loss (10% Non-Responding Nodes)

Computational Energy Loss Graph is shown in figure 9. This graph is used to represent extra energy loss due to extra computational in SEEDA scheme when 10% of nodes do not respond. In the graph the horizontal axes represent number rounds while the vertical axes represent computational energy loss in micro-joules. The graph shows that EDAS (blue line) scheme is more efficient in energy than the SEEDA

scheme (red line). This extra energy loss is the result of extra computation of cipher text c_j for all non-responding nodes at the level-2 Aggregators. This computation is not performed in EDAS. The IDs of non-responding sensors are produced and sent with aggregated result.

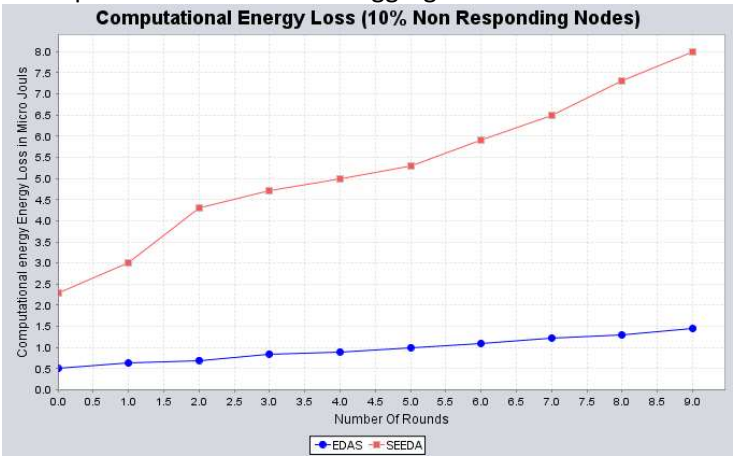

Figure 9: Computational Energy Loss (10% non-responding nodes)

6.3. Network Energy Consumed (10% Non-Responding Nodes)

This graph is shown in figure 10 and is used to represent energy consumption of the overall network when 10% of the sensors do not respond. In the graph the horizontal axes represent number rounds while the vertical axes represent energy consumed in milli-joules. The graph shows that EDAS scheme is more efficient than the SEEDA scheme from energy point of view. This extra energy consumption in SEEDA is because of the extra more than150 bits sent across the network due to non-responding nodes. According to the literature data sending consumes 70% energy of the sensor network. The red line in the plot is for SEEDA scheme while the blue line is for EDAS scheme.

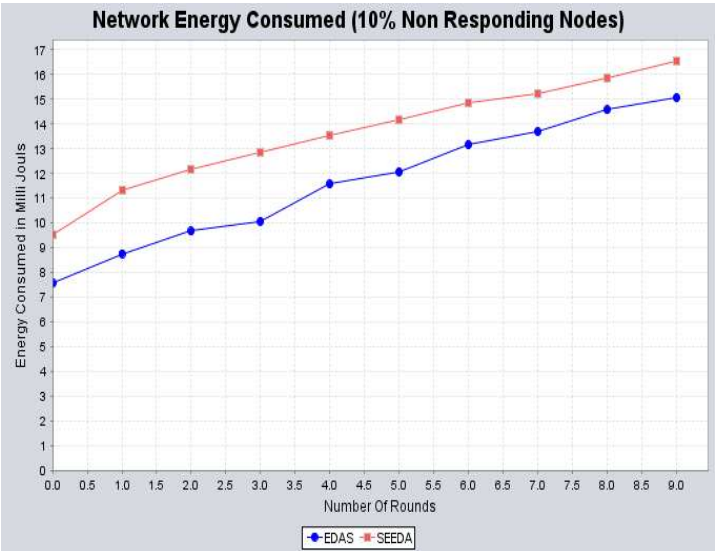

Figure 10: Network Energy Consumed (10% non-responding nodes)

6.4. Network Lifetime (10% Non-Responding Nodes)

This graph is used to represent overall lifetime of the network in seconds when 10% of sensors are non-responding and 90% are responding. The difference between lifetimes of EDAS (blue line) and SEEDA (red line) clearly shows that EDAS network lifetime is more by 90 seconds then corresponding SEEDA network. The vertical axes represent energy of the network consumed in milli-joule and the horizontal axes represents lifetime of the network in seconds. This time efficiency of EDAS is due to the fact that SEEDA has more computational and communication costs. This results in faster energy consumption of the SEEDA network, and it dies earlier. The graph is shown in figure 11.

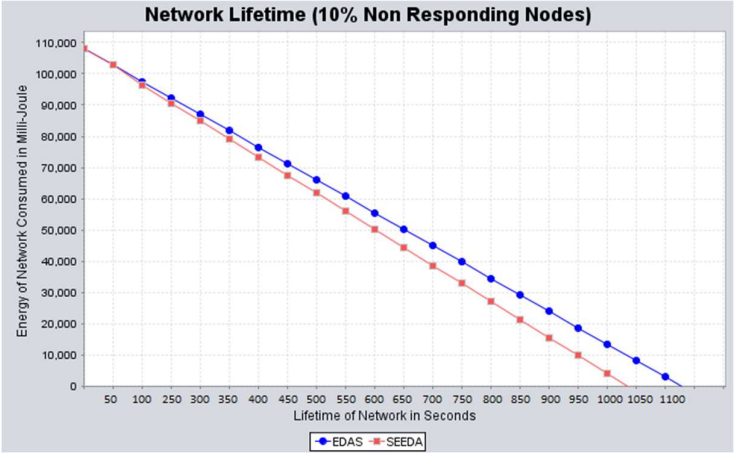

Figure 11: Network Lifetime (10% non-responding nodes)

6.5. Number of Bits Sent to the Base Station (30% Non-Responding Nodes)

This graph is shown in figure 12 and is used to represent number of bits sent to the base station when 30% of the sensors do not respond. In the graph the horizontal axes represent number rounds while the vertical axes represent number of bits/data sent to the base station. The graph shows that EDAS scheme is more than 400 bits efficient than the SEEDA scheme. These extra bits are generated in SEEDA scheme due to computation of cipher text for non-responding nodes. These extra bits/data will increase when the number of non-responding nodes exceed than 30%. The graph is shown in figure 6.5.

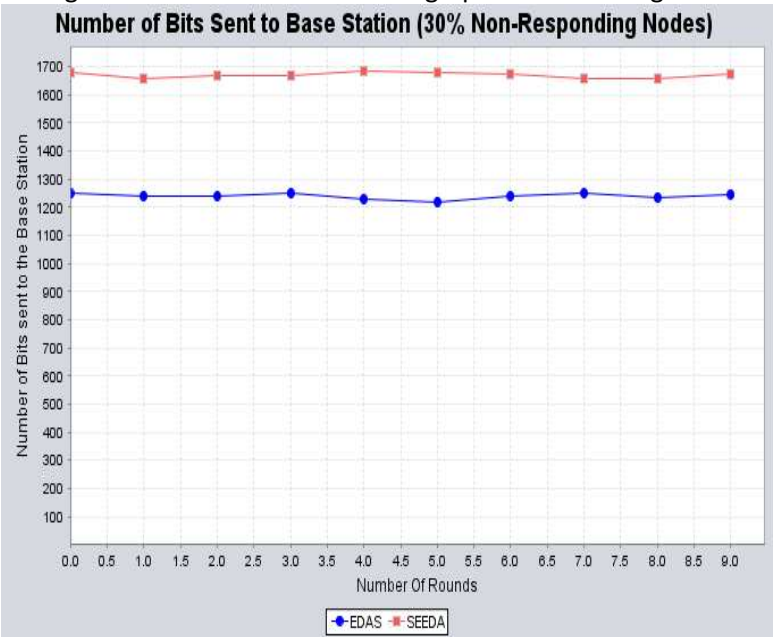

Figure 12: Number of bits sent to Base Station (30% non-responding nodes)

6.6. Computational Energy Loss (30% Non-Responding Nodes)

This graph is shown in figure 13 and is used to represent computational energy loss when 30% of the sensors are non-responding and 70% are responding sensors. In the plot, the horizontal axes represent number of rounds while the vertical axes represent computational energy loss in micro-joules. The graph shows that EDAS scheme is more efficient in computational energy than the SEEDA scheme. This extra energy loss is the result of extra computation for non-responding nodes at the level 2 Aggregators. The red line in the plot is for SEEDA scheme while the blue line is for EDAS scheme.

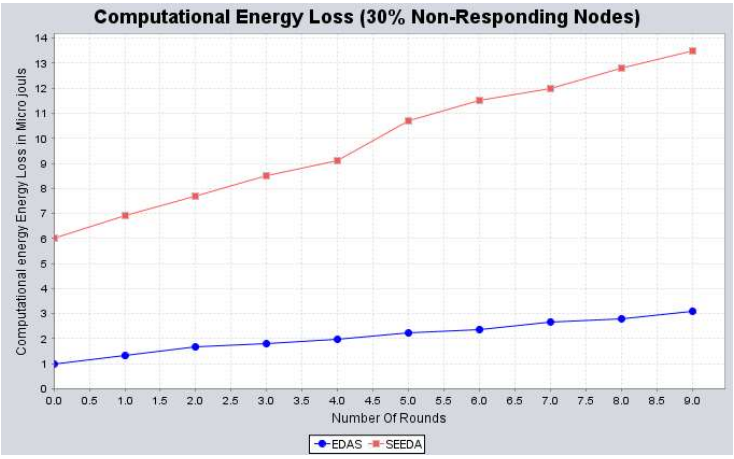

Figure 13: Computational Energy Loss Graph (30% non-responding nodes)

6.7. Network Energy Consumed (30% Non-Responding Nodes)

This graph is shown in figure 14 and is used to represent energy consumption of the overall network when 30% of the sensors do not respond. In the graph the horizontal axes represent number rounds while the vertical axes represent energy consumed in milli-joules. The graph shows that EDAS scheme is more efficient than the SEEDA scheme from energy point of view. This extra energy consumption in SEEDA scheme is due to the fact that more than 400 bits are sent across the network for non-responding nodes. Data transmission consumes 70% energy of the sensor network. The red line in the plot is for SEEDA scheme while the blue line is for EDAS scheme.

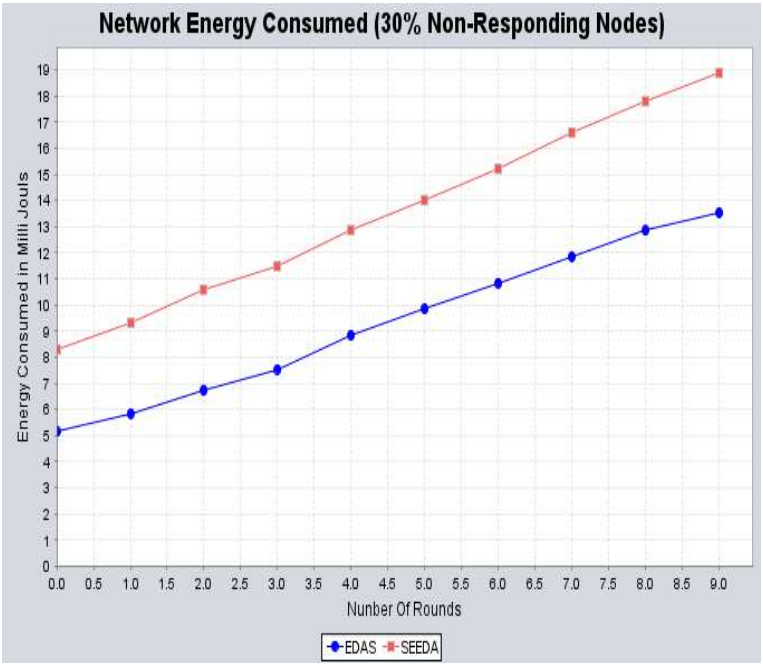

Figure 14: Network Energy Consumed (30% non-responding nodes)

6.8. Network Lifetime (30% Non-Responding Nodes)

This graph is used to represent overall lifetime of the network in seconds when 30% of sensors are non-responding. The difference between lifetime of EDAS (blue line) and SEEDA (red line) clearly shows that the lifetime of EDAS scheme is more by 270 seconds than the SEEDA scheme. The vertical axes represent energy of the network consumed in milli-joule and the horizontal axes represents lifetime of the network in seconds. This time efficiency of EDAS is due to the fact that SEEDA has more computational and communication costs. This results in faster energy consumption of the SEEDA network and it dies earlier.

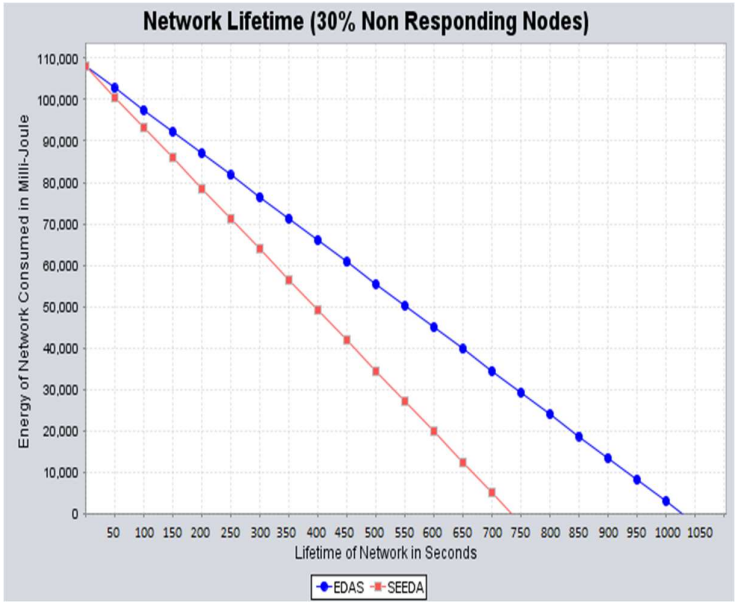

Figure 15: Network Lifetime (30% non-responding nodes)

6.9. Latency / Data Delay (10% - 70%)

Data takes time when sent from sensor node to the base station. This delay of data from sensors to base station is called Latency. The latency of SEEDA and EDAS are shown in figure 16. A vertical axes represents time delay in milli-seconds per round. The horizontal axes represent number of non-responding nodes in percentage (from 10% to 70%). The red line in the plot is for SEEDA scheme while the blue line is for EDAS scheme. The graph clearly shows that SEEDA protocol has lager data delay than EDAS. This is due to the fact that SEEDA has higher computational cost. Higher computational cost results in lager delays to data.

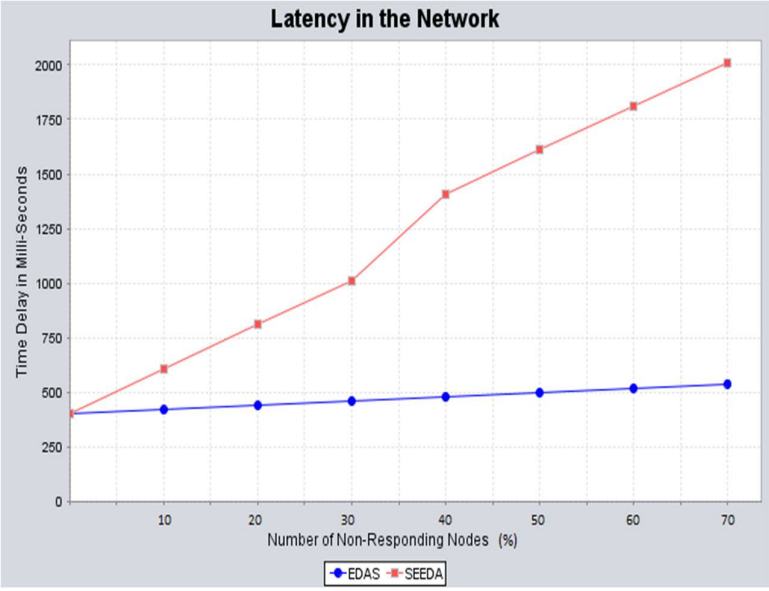

Figure 16: Latency / Data Delay (10% - 70% Non-Responding Nodes)

6.10. Data Accuracy (10% – 70% Non-Responding Nodes)

Data Accuracy of the EDAS and SEEDA schemes are shown in figure 17. In the graph the vertical axes represent data accuracy which has standard value of 1. The horizontal axes represent number of non-responding nodes in percentage (10% to 70%). The blue line in the plot represents EDAS scheme while the red represents SEEDA scheme. It clears from the graph that as numbers as of non-responding increase the accuracy of both the schemes decreases. But the accuracy of SEEDA is less than that of EDAS scheme.

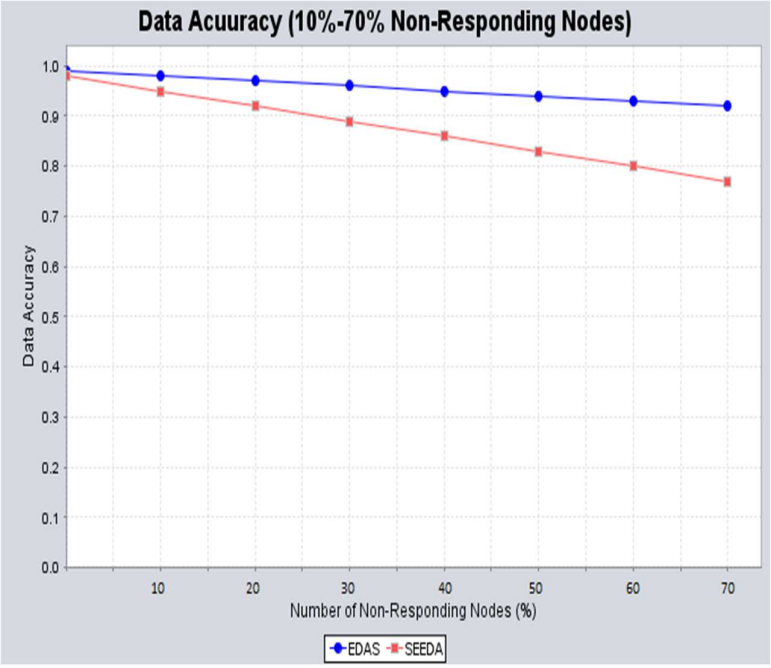

Figure 17: Data Accuracy (10% - 70%)

7. Conclusions

In the paper a new scheme, Efficient Data Aggregation Scheme in Secure Tree Based Wireless Sensor Network (EDAS) is proposed to achieve secure yet end-to-end data aggregation. The proposed scheme ensures end-to-end privacy of data and lesser number of bits is transmitted from sensors to base station as compared to Secure End-to-End Data Aggregation (SEEDA) scheme. The EDAS protocol adopts the best feature of hop-by-hop data aggregation i.e., transmission of less number of bits/data from sensors to base station. EDAS also adopts the best characteristic of end-to-end protocols that is end-to-end data privacy. EDAS protocol is also more efficient in energy consumption than SEEDA. Again, EDAS is also more efficient in computational energy cost than that of corresponding SEEDA scheme. One unique feature is that the efficiency of EDAS increases as the number of non-responding nodes in network increases. Simulation results shows that the proposed scheme is more efficient in terms data accuracy. My proposed scheme has smaller latency/data delay than the existing scheme. The lifetime of the EDAS network is more than that of SEEDA scheme.

References

[1] K. Akkaya, M. Younis, “A survey on routing protocols for wireless sensor networks” International Journal on Ad Hoc Networks 3,pp. 325–349, 2005.

[2] J. Yick, B. Mukherjee, D. Ghosal, “Wireless Networks Survey”, The International Journal of Computer and Telecommunications Networking, vol. 52, pp. 2292-2330, 2008.

[3] Sinha, A. Chandrakasan, “Dynamic power management in wireless sensor networks”, IEEE Design and Test of Computers 18 (2), pp. 62–74, 2001.

[4] Chandrakasan, R. Amirtharajah, S. Cho, J. Goodman, “Design considerations for distributed micro sensor systems”, Proceedings of IEEE Customs Integrated Circuits Conference (CICC), pp. 279–286, May 1999.

[5] Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, “SPINS: Security protocols for sensor network”, Journalof Wireless Networks vol. 8, no. 5, pp. 521–534, 2002.

[6] Krishnamachari, D. Estrin, S. Wicker, “The impact of data aggregation in wireless sensor networks”, Proceedings of the 22nd International Conference on Distributed Computing Systems Workshops, pp. 1-4, 2002.

[7] Castelluccia, E. Mykletun, G. Tsudik, “Efficient Aggregation of encrypted data in wireless sensor networks”, Transactions of MobiQuitous, pp. 1-9, 2005.

[8] Castelluccia, C-F Chan, E. Mykletun, G. Tsudik, “Efficient Provably Secure Aggregation of encrypted data in wireless sensor networks”, ACM Transactions on Sensor Networks, vol. 5, pp. 1-36, May 2009.

[9] A.S. Poornima, B.B. Amberker, “SEEDA: Secure end-to-end data aggregation in Wireless Sensor Networks”, In Proceeding of the 7th International Conference of Wireless and Optical Communications Networks (WOCN), pp. 1-5, 2010.

[10] H. Cam, S. Ozdemir, P. Nair, and D. Muthuavinashiappan, “ESPDA: energy-efficient and secure pattern-based data aggregation for wireless sensor networks”, IEEE Sensors–The Second IEEE Conference on Sensors, pp. 446-455, Oct. 2003.

[11] H. Sanli, S. Ozdemir, and H. Cam, “SRDA: Secure Reference-Based Data Aggregation Protocol for Wireless Sensor Networks”, Proc. IEEE 60th Int’l Conf. Vehicular Technology (VTC 04-Fall), vol. 7, pp. 4650-4654, Sept. 2004.

[12] Y. Yang, X. Wang, S. Zhu, and G. Cao, “SDAP: A Secure Hop-by- Hop Data Aggregation Protocol for Sensor Networks”, ACM Trans. Information and System Security (TISSEC), vol. 11, pp. 1-43, 2008.

[13] Westhoff, J. Girao, and M. Acharya, “Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation”, IEEE Trans. Mobile Computing, vol. 5, pp. 1417-1431, Oct. 2006.

[14] S. Ozdemir, “Concealed Data Aggregation in Heterogeneous Sensor Networks Using Privacy Homomorphism”, Proc. IEEE Int’l Conf. Pervasive Services, pp. 165-168, July 2007.

[15] S. Ozdemir and Y. Xiao "Hierarchical concealed data aggregation for wireless sensor networks", In: Proceedings of the Embedded Systems and Communications Security Workshop in conjunction with IEEE (SRDS), pp. 1-5, 2009.

[16] Josna J, Manoj Kumar S, Joyce J, “Energy Efficient Recoverable Concealed Data Aggregation in Wireless Sensor Networks”, IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), pp.322-329, 2013.

[17] Joyce J, M. Princy, Josna J, “PEPPDA: Power Efficient Privacy Preserving Data Aggregation for Wireless Sensor Networks”, IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), pp.330-336, 2013.

[18] Abhilash L N, D. Goenka, C. Kumar, “Dynamic Data Aggregation for Energy Optimization in Multi-Hop Wireless Sensor Networks”, IEEE International Advance Computing Conference (IACC), pp. 143-148, 2014.