IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 5, Ver. II (Sept - Oct. 2016), PP 23-33
www.iosrjournals.org

Secure Strategy for Key Pre-Distribution in Wireless Sensor Networks

D.S.S.L.Sruthi , D.Sesha Manoj, I.Ravi Prakash Reddy
Department of Information Technology, G.Narayanamma Institute of Science & Technology, Hyderabad

***Abstract**: Wireless sensor networks (WSN) are a collection of sensor nodes, which is deployed spatially among various environments. WSNs are deployed in environments where wired networks are difficult to set up. They are mainly used to record sensitive information regarding physical or environmental conditions. As they collect sensitive information, secure and accurate transmission of data is important. For secure transmission of data, key management is a corner stone in wireless sensor networks.*

Wireless sensor networks are infrastructure less. Due to this reason, key distribution among increasing number of nodes by third parties is a challenging problem. Hence, symmetric key pre-distribution algorithms are introduced for key distribution. However, most schemes suffer from low scalability, high storage overhead and it is difficult to implement security with existing solutions. In order to provide good key sharing and secure connectivity, an asymmetric algorithm called elliptical curve cryptography (ECC) will be used. After collecting data from deployed environment, sensor nodes will send data to sink node, which receives data and process the information. Data will be routed to sink node in secure environment using shortest path routers. The schemes will be implemented in java programming language and results would be derived to show/compared for more efficiency in terms of secure connectivity, storage overhead, computational cost and communication cost.

I. Introduction

Wireless sensor networks are comprised of small electronic devices, which exchange information by interacting with each other. These electronic devices consist of data processing unit, memory unit, energy resource and transceiver. These are widely used in commercial and industrial areas such as environmental monitoring, health care, and process monitoring. Location and positioning information can also be obtained through the global positioning system (GPS) or local positioning algorithms. After sensing information, this information is sent to the sink node. Sink node is a sensor node, which collects information from all devices and processes the data. WSNs work with some protocols and algorithms with self organizing capabilities. Routing algorithms are used to send data from source to destination.

As WSNs record sensitive information, secure transmission is important. Hence, key pre-distribution is introduced using asymmetric key algorithms. Here keys are distributed after the nodes are deployed. After generation of keys to sensor nodes, they identify common key with neighboring nodes and establish connection in between the nodes. If the nodes have no common key, then they establish path through successive secure links.

1.1 Classifications

WSNs are classified into three categories: Proactive, Reactive, Hybrid Proactive provides the information of relevant parameters at regular intervals. Reactive provides information immediately after the event had occurred. They are used in collecting sensitive information. Hybrid is the combination of both proactive and reactive sensors.

1.2 Characteristics

The main characteristics of WSNs are mobility of nodes, limited bandwidth, dynamically varying network topologies and limited resources.

- Ability to cope up with node failure.
- Mobility of nodes
- Dynamic network topology
- Scalability to large scale of deployment.
- Ability to with-stand harsh environmental conditions.
- Limited battery power.
- Self organized.

1.3 Applications

The wireless sensor networks were primarily developed as military applications in battlefield. Now, WSNs are used in many other applications like industries and healthcare. Commercial and industrial applications like monitoring hardware equipment or old buildings, where it is difficult to attach wired sensors, there WSNs are used.

Area Monitoring: Monitoring is a common application of WSNs. In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. A military example is the use of sensors to detect enemy intrusion.

Air Pollution Monitoring: Wireless sensor networks have been deployed in several cities to monitor the concentration of dangerous gases for citizens living in cities.

Forest Fires Detection: A network of sensor nodes can be installed in a forest to control when a fire has started. They will be equipped with sensors to control temperature, humidity and gases which are produced by fire in the trees or vegetation.

***Health care Monitoring*:** Here, sensors are fixed to patient's body and collect information regarding heart-beat, pulse rate to monitor patient's health condition.

Search and Rescue Operations: Communication in locales where there is no help from wireless infrastructure.

1.4 Security Attacks

Security provides confidentiality, authentication, non-repudiation, and integrity to the confidential data. Security attackers can be classified in two types: Passive attack, Active attack.

***Passive Attack* –** A Passive attack monitors unencrypted traffic and looks for clear-text passwords and sensitive information that can be used in other types of attacks. Passive attacks include traffic analysis, monitoring or unprotected communications, decrypting weakly encrypted traffic, and capturing authentication information such as passwords.

***Active Attack* –** This type of attackers is attempting to make changes to data, tries to bypass or break into secured system. This can be done through stealth, viruses, worms, or Trojan horses.

W*ormhole Attack* – Wormhole nodes make a fake a route that is shorter than the original one wit in the network. This can confuse routing mechanisms which rely on the knowledge about distance between nodes. It has one more malicious nodes and a tunnel between them. The attacking node captures the packets from one location and transmits them to other distant located node which distributes them locally.

II. Literature Survey

A Wireless Sensor network (WSN) consists of a large number of spatially distributed tiny sensor devices which are also known as sensor nodes. WSNs have some unique features, for instance, limited power, ability to withstand harsh environmental conditions, ability to cope with node failures, mobility of nodes, dynamic network topology, communication failures, heterogeneity of nodes, large scale of deployment and unattended operation.

Hence, for secure routing of data, appropriate key management also plays major role. Some symmetric key management techniques are introduced. But they have certain drawbacks regarding some performance issues. Hence, a new unital based key management scheme for secure connectivity and key sharing is introduced. As these performance metrics depends on network size, they are analyzed with total network size.

A new mechanism for key establishment using pre-distribution of random set of keys to each node has been proposed [4]. Nodes are randomly deployed to ensure secure connectivity. When active link is compromised or any node is deleted from the network, it should take alternate measure for transmission of data and for secure connectivity keys must be distributed equally among all nodes [5]. Security must be ensured when dealing with large scale of sensor networks as there is chance for unauthorized nodes to enter into the environment [9].

2.1 Data Transmission Protocols

The multipath transmission protocols are divided into two categories. For permitting spontaneous, infrastructure-less networking and efficient back-to-back correspondence with the nodes in a network, a routing protocol is put to use for discovering maximum number of routes among the nodes.

- Homogeneous sensor network.
- Heterogeneous sensor network.

2.2 Aggregation of Encrypted Data

Wireless sensor networks (WSNs) are ad-hoc networks consists of tiny devices with little computation and energy capacities. For such devices, data transmission is a high energy-consuming operation. Hence, it becomes essential for the lifetime of WSN to minimize the number of bits sent by each device. One approach is to aggregate sensor data (adding) along the path from sensors to the destination. Aggregation becomes especially challenging if privacy between sensors and the sink is required. Encrypted data is seen as points on elliptic curve, so that, original message is not known at intermediate nodes.

2.3 Related Work

Here, key pre-distribution protocols through symmetric key pre-distribution. Earlier they showed some benefits regarding scalability, network connectivity and security. In deterministic schemes, secure connectivity is achieved, while in probabilistic schemes, secure connectivity is not achieved, because they are obtained by the shared keys from node to node.

Probabilistic schemes: In probabilistic technique, two neighboring nodes can have secure link with certain probability. If secure link is not found, connection is established through successive secure links. Eschenauer and gligor proposed basic random key pre-distribution scheme denoted by RKP. In this scheme, each node is loaded with ring of k keys from large pool s of keys. After completion of the deployment step, each node i exchange with its neighboring nodes j. key identifiers are maintained at each node, so as node j can identify keys that share with node i. If two neighboring nodes share one key, then secure link is established and can compute their secret key which is common key. Otherwise, path is established by successive secure links. Key ring size k and key pool size s are taken where the intersection of two rings of keys are not empty [3].
Chan et al. introduced protocol called Q - Composite scheme, which improves resilience of RKP. In this scheme, two neighboring nodes can have secure link only if they share at least Q keys. Pair wise secret keys are compared by the hash of shared keys. He also proposed perfect secure key distribution. Before deployment nodes are pre loaded with Pc x n keys. Here the drawback is non scalability [4].

***Deterministic schemes*:** Deterministic schemes ensure secure connectivity. Many solutions are given to this approach. A deterministic key pre-distribution is obtained by giving each link(i,j) a distinct key Ki,j. Choi et al. introduced an approach for storing (n+1)/2 keys at each node. He proposed hash function based key establishment to store half of the symmetric keys while processing remaining half at every node.

Another approach named LEAP (Localized Encryption and Authentication Protocol) which uses common transitory key preloaded into nodes at the time of deployment. Transitory key is used to create pair wise session keys and they are cleared from memory of nodes after deployment. LEAP algorithm is based on assumption that at time T_{min} sensor node is secure and cannot be compromised [9].

The combinatorial design for key pre-distribution is proposed by Camtepe and Yener. They proposed new deterministic scheme depends on symmetric balanced incomplete block design (SBIBD). This technique maps from SBIBD to key pre-distribution which is used to construct m^2+m+1key rings from key pool s and each ring has k=m+1 keys and two rings share one common key. Here main advantage in this scheme is secure connectivity. SBIBD and trade based techniques achieve scalability up to O (K^4) [11].

***Asymmetric Key Cryptography*:** Cryptographic techniques are applied to protect communication in wired and wireless networks. Cryptography is a study of applied mathematical techniques with regard to the security process of transmission as well as storage of information. The original message (unencrypted text) is also known as plaintext; whereas concealed version is named cipher text. Transforming the plaintext to the cipher text is named encryption and method of converting the cipher text into a plaintext is called decryption. The cipher text is converted into the plaintext for verifying the sender's identity, the integrity of data, or some combination. The keys manage the process of encryption and decryption, when the amount of information is less and is used by the cryptographic algorithms. Asymmetric key cryptography, otherwise called public key cryptography, uses a pair of mathematically connected cryptographic keys, called public key and private key, for encryption/decryption and for signing/verification of information.

Unital design: It is required to build smart techniques to create block of keys on nodes for securing network links. But in existing solutions, design of key rings suffers from low scalability. This motivates building of new unital design theory.

New key pre-distribution scheme: Here, before deployment stage, blocks of m order are generated. Here each node is pre loaded with t completely disjoint sets, t is protocol parameter. If two nodes share more keys, then pair wise secret keys are hashed and common keys are concatenated to each other.
By using t-UKP scheme, storage overhead is reduced. It is calculated by l*t*(m+1).Unital based scheme UKP* gives equal key sharing and better results are equal to 1.
Nodes are given with unique identification number which is homogeneous in functionalities. Then keys are distributed to each and every node. Now, selection of source and destination takes place. Data is forwarded to

destination and then to sink node. Data is forwarded from source to destination through shortest path algorithm. If attacker is found in the middle of forwarding path, then data resolves the path and takes another route for secure data transmission.

III. Methodology

Existing random key pre-distribution scheme requires a large memory space to store the key ring.Q-composite scheme approach degrades the network secure connectivity. Perfect secure pair wise key pre-distribution scheme. Drawback of this scheme is the non scalability, because the number of the stored keys depends linearly on the network size. A naive deterministic key pre-distribution scheme approach allows to reduce the required stored keys to the half of network size, however, it is obvious that this scheme remains non scalable enough. These systems have drawbacks like lack of network scalability, more storage overhead, more energy consumption, insecure connectivity.

3.1 Proposed System

An enhanced algorithm in key pre-distribution for WSNs is proposed. The basic mapping from unitals to key pre-distribution gives better sharing of keys among all nodes. Elliptical curve cryptography is introduced for secure connectivity and less storage overhead. This system has advantages like network scalability is high, less storage overhead, less energy consumption, efficient secure connectivity.

Nodes are deployed in random manner. Nodes are given with unique identification number which is homogeneous in functionalities. These nodes capture the data that exists in the environment and send the data to the base station. They interact with each other and send data to the base station. Keys are distributed prior to the communication.

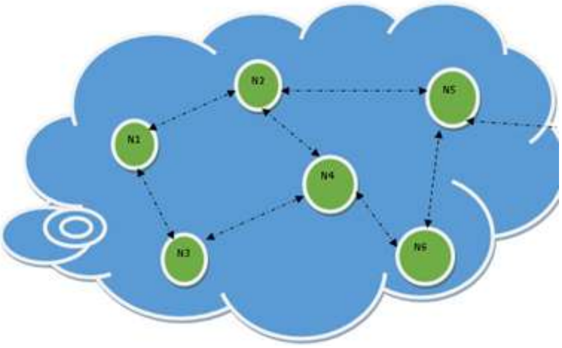


Figure 3.1: System Model for WSN

In the above figure, it is shown that nodes are deployed in random manner. N1, N2, N3, N4, N5 are sensor nodes that are deployed. In the proposed system, elliptical curve cryptography is used. ECC algorithm is based on algebraic features. Point multiplications in ecc algorithm are important because sensors need to process information quickly, as point multiplications are used for fast calculations. Breaking of algorithm is difficult for attacker as it strongly depends on the complexity of the elliptic curve in ecc.

3.2 Key Management

For routing data, key management has to be done. Key pre-distribution is done by initially generating key pairs.

Generation of key pairs: Elliptical curve cryptography provides key exchanges between communicating nodes. First for the generation of key pairs, we have to figure public key and private key. Security of the proposed algorithm depends on complexity of elliptic curve. Hence, we take points on finite field fp. Order of the curve is also considered.

Generation of private key: For getting private key, choose random integer dA , where 0< d <n , n is the order of curve.

Generation of public key: To get public key QA, we have to use scalar point multiplication, *QA= dA*G.* Here 'G' is the generator point. And here the keys are not exchangeable, because dA (private key) is an integer and QA is point on the curve. These are defined over the curve $y^2 = x^3 + ax + b$, and defined over $4a^3 + 27b^2 \neq 0$, where G, a, b are domain parameters, which are pre defined.

3.3 Encryption

For encryption, after generating public key QA, choose a random number r, where 0<r<n. Then find an appropriate point 'R', where R=r*G. And then we have to calculate symmetric key 'S'. S=r.QA. Then R is transmitted from S symmetric key from which it is encrypted.

3.4 Decryption

Here in decryption, we have to decrypt with the symmetric key and get value of R with plain text.

- S=dA*R
- S=dA*R=dA*r*G=r*(dA*G)=r.QA

ECDSA (Elliptical Curve Digital Signature Algorithm) is used for signature generation. Here for signing a message m by sender A, using A's private key d_A and public key $Q_A = d_A * G$ Calculate e = hash (m), where hash is a cryptographic hash function, such as SHA. And in decryption it is reversed.

3.5 Key Distribution

- $(Q_{A,}d_A)$ – Public, Private Key pair
- $(Q_{B,}d_B)$ – Public, Private Key pair

1. The end A computes $K = (x_K, y_K) = d_A * Q_B$
2. And at receiver end $L = (x_L, y_L) = d_B * Q_A$
3. Since $d_AQ_B = d_Ad_B G = d_Bd_AG = d_BQ_A$.
4. Therefore K = L and Hence, $x_K = x_L$
5. Like the secret key is shared.

ECC algorithm can transfer keys in faster manner.

Procedure

Step 1: First enter number of nodes to be inserted in environment.

Step 2: Next, keys are distributed to all the nodes. Public key and private keys are transmitted i.e., pk and sk are transmitted to each node. Keys are exchanged skA and pkA are one pair, skB and pkB are another pair (skA ,pkA) (skB ,pkB)

Step 3: Now, after distributing keys to each node, select a node to send and another node to receive.

Step 4: Then enter message that is to be sent from source to destination. Message is encrypted using key and it is sent along with a large random point R. *R=r*G*

Step 5: At the receiving side, to get plain text, we have to decrypt with the symmetric key to get the value of point R. Through that point, 'R' original message will appear by using secret key.

Step 6: After receiving at destination node, the data is sent to the sink node using shortest Path

3.6 Shortest Path Calculation

In our proposed system, after key pre-distribution, routing takes place. This shortest path routing is calculated by distance vector algorithm. Initially distances are stored in arrays and minimum distances are calculated. First node is taken and it was later incremented. Number of nodes and first node are taken in loop and calculate the distance for first node to all other nodes. Initially, two nodes are taken and the distance is calculated. The result is stores in another array. Next, the immediate node occupies first element in array and calculates distance with the other node. Size of the path is calculated. Comparison is done for minimum and maximum distances. Minimum valued nodes are stored in one array and neighbor nodes are stored in another path array.

Data transmission from Source to destination

After key pre-distribution, source and destination are selected. Then data is sent from source to destination through the shortest path method. At the intermediate nodes, only encrypted data is seen and original data is appeared only at destination node.

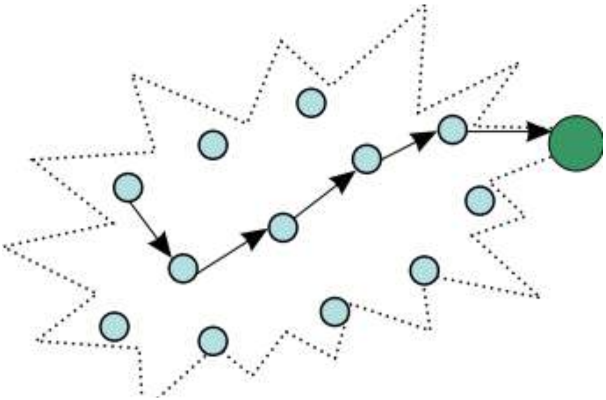


Figure 3.2: Data Transmission from Source to Destination

Data transmission for sink node

After receiving data to the destination node, sink node is selected and data from all sensor nodes are sent to sink node. In the sink node information is processed and is sent to respective base station.

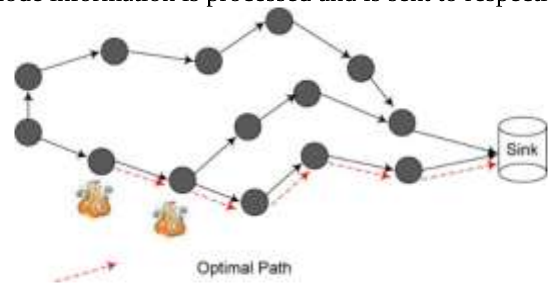


Figure3.3: Data Transmission from Destination to Sink Node

3.7 Functional Requirements

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.

Input: Deployment of network with a specific number of node.

Output: Sensing data transform to the sink node.

Internal Process: Sensor nodes sensing data and transform to the destination. Then data from destination aggregates and transform to the sink node.

3.8 Non-Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions In general; functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be.

Non-functional requirements can be divided into two main categories:

Execution qualities such as security and usability, which are observable at run time.

Evolution qualities such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

The major non-functional Requirements of the system are as follows

Usability: The system is designed with completely automated process. Hence there is no or less user intervention.

Reliability: The system is more reliable because of the qualities that are inherited from the chosen platform java. The code built by using java is more reliable.

Performance: This system is developing in the high level languages and using the advanced front-end and back-end technologies it will give response to the end user on client system with in very less time.

Supportability: The system is designed to be the cross platform supportable. The system is supported on a wide range of hardware and any software platform, which is having JVM, built into the system.

3.9 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. Three key considerations involved in the feasibility analysis are Economical Feasibility, Technical Feasibility, Social Feasibility.

IV. Design of UML Diagrams

4.1 Usecase Diagram

Actor: Sensor node

Usecase:

- ❖ Init → Initialization of sensor nodes
- ❖ KeyAgree →Key agreement is done by using elliptic curve cryptography algorithm
- ❖ Read Data → After initialization, the sensor node is ready to send data
- ❖ Find Route to Sink → Distance vector routing are used to find the route to the sink.
- ❖ Send Data → Data is send from source to the given destination.

Actor : Sink

Usecase :

- Init() → Initialization of the Sink.
- Key Agree() → Key agreement is done by using elliptic curve cryptography Algorithm
- Send Data → Data is receive from source to the sink.

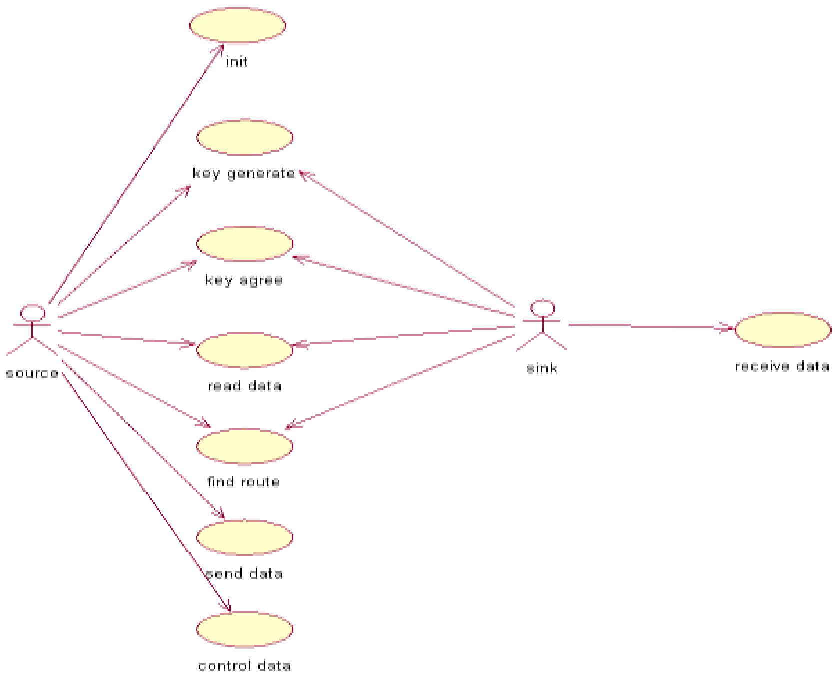


Figure4.1 : Usecase Diagram

4.2 Sequence Diagram:

When the sensor node senses the data that data is sent to the neighbor sensor node, then that sensor node will sent to its neighbor towards the sink.

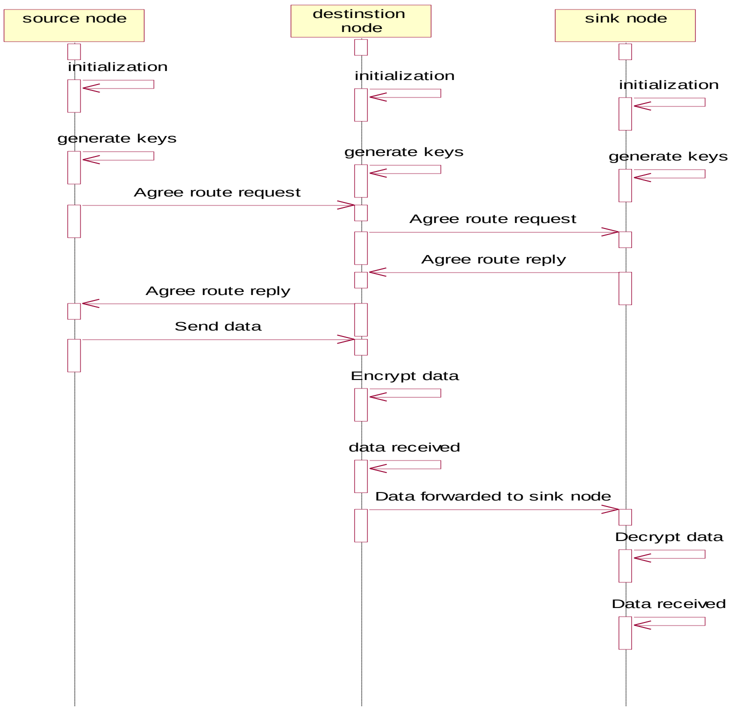


Figure 4.2: Sequence diagram

4.3 Activity Diagram

In activity diagram, connection between flow of events is shown. First, in source node, destination node and sink node keys are generated and then data is forwarded to destination node, after key verification. Then data is directed to the sink node. In activity diagram, decision controls are there for the verification of various conditions. Activity diagram depicts about various events occurring in the system

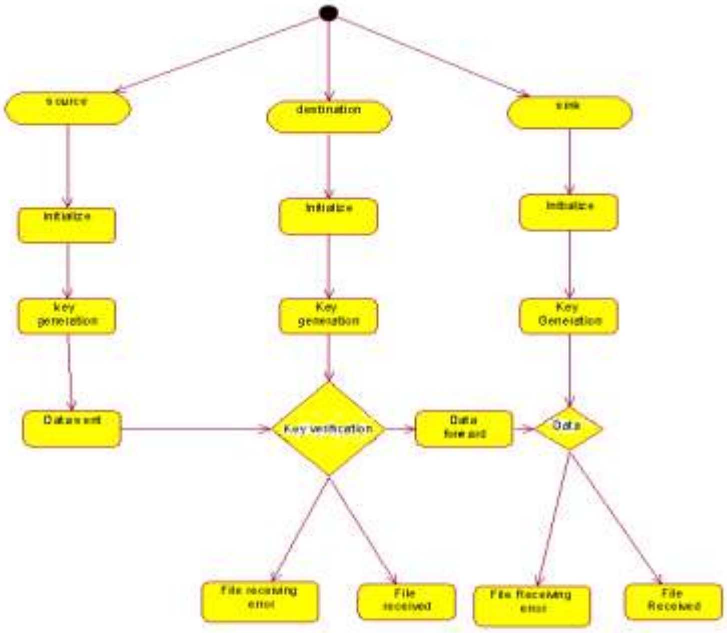


Figure 4.3: Activity Diagram

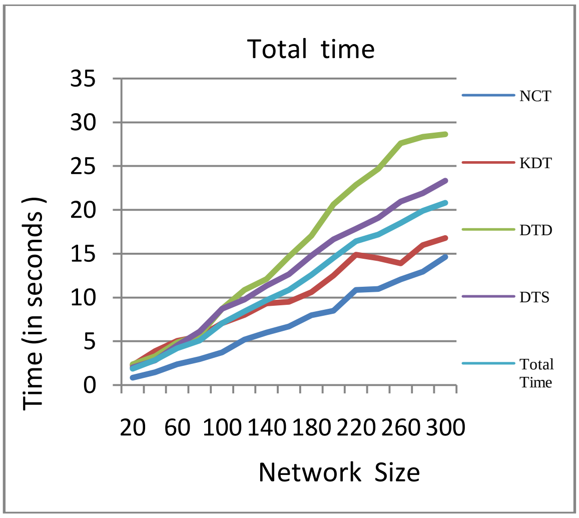
V. Results

5.1 Time Complexity

Time complexity is obtained by considering network creation time (NCT), Key distribution time (KDT), data transmission to destination (DTD) and data transmission to sink node (DTS).

$$\sum(NCT+KDT+DTD+DTS)$$

When the network size is increased to 300, the total time becomes 20.8 seconds. It is taken for average of four program runs.

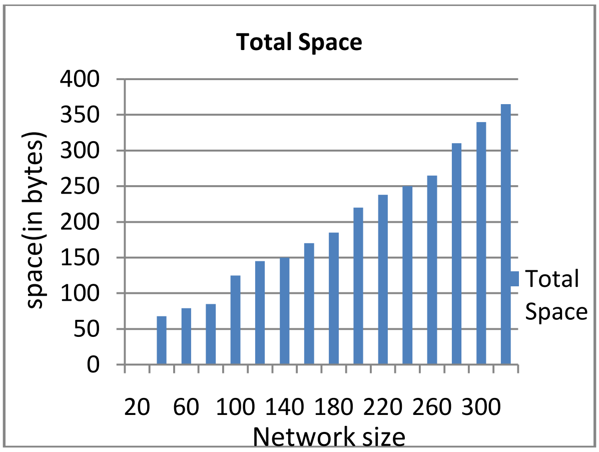


5.2 Space Complexity

Space complexity is calculated by considering neighbor nodes (NN) and data stored at destination node and data stored at sink node.

$$\sum(NN+DN+SN)$$

When the network size is increased to 300, the total space becomes 365 bytes. It is taken for average of four program runs.

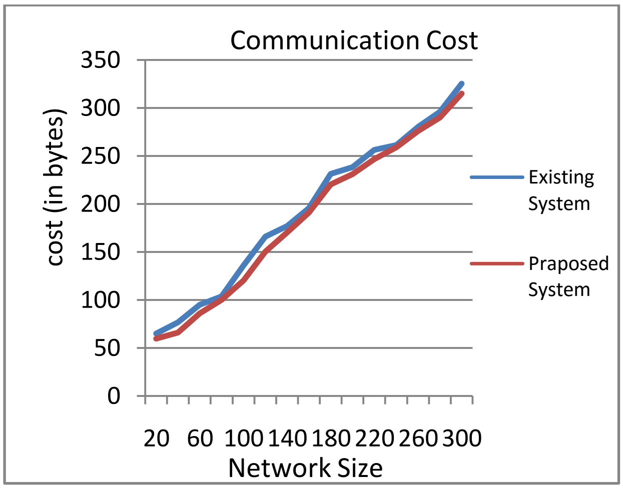


5.3 Communication Cost

It is defined by key size, encrypted data size, message size and total number of nodes in the path.

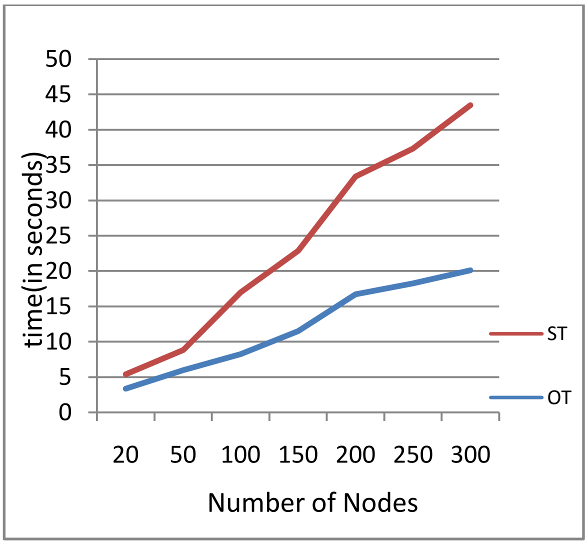
$$\sum(ECC\_SIZE+ENC\_SIZE+MSG\_SIZE)*(N+1)$$

ECC_SIZE is key distribution time, ENC_SIZE is encrypted data size, message digest size and N is the total number of nodes in path.



5.4 Secure Connectivity

It is estimated by taking attacker into consideration. Here, attacker is found in the middle of path. Then data reaches to the destination by secure routing. Here, secure transmission of data is plotted by comparing the paths of original transmission and secure transmission.

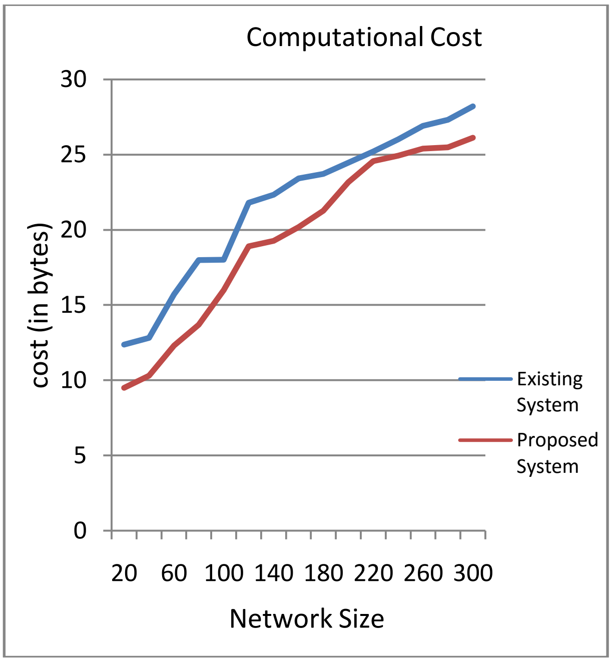


5.5 Computational Cost

Computational cost is calculated by considering the key generation time, encryption time, message attributes.

$$\sum(\mathit{ECC\_time}+\mathit{ENC\_time}+\mathit{MSG\_SIZE})*(S+R)$$

Where ECC_time is the time for generating the key to encrypt the data, ENC_time is encryption time for attributes, MSG_SIZE is for generating attributes for the message.



VI. Conclusions and Future Work

Asymmetric key management technique, ECC is used for secure connectivity and key sharing, which overcomes the drawback that exists in symmetric key algorithms. And data transmission in secure manner is achieved. Secure connectivity and network resiliency is analyzed against existing systems and have given better performances. Computation cost is reduced as proposed algorithm ECC uses point multiplications which computes faster. Time complexity, space complexity, communication cost are analyzed with equal network size. Future Work can be extended in the case of key distribution at increasing number of nodes and in the case of storage memory space in each sensor node. Security can be increased to sensitive information.

References

[1]. W. B.kit, Y.Challal, A. Bouabdallah, V. Tarokh, “A Highly scalable Key Pre-Distribution Scheme for Wireless Sensor Networks”, IEEE Transaction on Wireless Communications, vol No. 2, PP.948-959 February 2013.

[2]. Y. Zhou, Y. Fang, and Y. Zhang, “Securing Wireless Sensor Networks: a Survey,” IEEE Communication Survey, vol. 10,no.1–4,pp. 6–28, 2008.

[3]. L. Eschenauer and V. D. Gligor, “A key-Management Scheme for the Distributed Sensor Networks,” IEEE Transaction on Wireless Communications, vol-1, pp. 41–47, 2002.

[4]. H. Chan, A. Perrig, and D. Song, “Random Key Pre-distributions Schemes for Sensor Networks,” in Springer Publishers, vol-3, pp. 197–213, 2003.

[5]. W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, “ A Key Pre-Management Scheme for Wireless Sensor Networks using the Deployment Knowledge,” IEEE iConference, pp. 586–597,2004.

[6]. C. Castelluccia and A. Spognardi, “A Robust Key Pre-Distribution Protocol for Multi-phase Wireless Sensor Networks,” IEEE Secure communications, vol-2, pp.351–360, 2007.

[7]. D. Liu and P. Ning, “Establishing Pairwise Keys in Distributed Sensor Networks,” ACM Computer and Communications Society , pp.52–61, 2003.

[8]. Z. Yu and Y. Guan, “A Robust Group-Based Key Management Scheme for Wireless Sensor Networks,” IEEE Transaction on Wireless Communications, vol-2, pp. 1915- 1920, 2005.

[9]. S. Ruj, A. Nayak, and I. Stojmenovic, “Fully Secure Pairwise and Triple Key Distribution in Wireless Sensor Networks using the Combinatorial Designs,” IEEE Information communications, pp. 326–330, 2011.

[10]. S. Zhu, S. Setia, and S. Jajodia, “Leap: Efficient Security Mechanisms for Large-scale Distributed Sensor Networks,” Conference on Computers and Communication Security, pp. 62– 72, 2003.

[11]. S. A. C¸ amtepe and B. Yener, “Combinatorial Design of key distribution Mechanisms for Wireless Sensor Networks,” IEEE/ACM networks, vol.15, pp. 346–358, 2007.

[12]. A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, and J. D. Tygar, “Spins: Security Protocols for Sensor Networks,” ACM Mobile Communications vol- 2, pp.189- 199, 2001.

[13]. B. Maala, Y. Challal, and A. Bouabdallah, "Hero: Hierarchical Key Management Protocol for Heterogeneous WSN," The International Federation for Information Processing and Wireless Sensor and Actor Networks vol-1, pp. 125–136, 2008.

[14]. W. Bechkit, Y. Challal, and A. Bouabdallah, "A New Scalable Key Predistributions Scheme for WSN," in Proc, IEEE International Conference on Communications, pp. 1–7, 2012.

[15]. J. Zhang and V. Varadharajan, "Wireless Sensor Network Key Management Survey and Taxonomy," J. Network Computer Applications., vol. 33, no. 2, pp. 63–75, 2010.

[16]. S. A. C¸ amtepe and B. Yener, "Key Distribution Mechanisms for Wireless Sensor Networks: a survey," Technical Report TR-05-07, 2005.

[17]. R. Blom, "An Optimal Class of Symmetric Key Generation Systems," Pro Eurocrypt Workshop Advances Cryptology: Theory Application Cryptographic Techniques, pp. 335–338, 1985.

[18]. T. Choi, H. B. Acharya, and M. G. Gouda, "The best keying protocol for Sensor networks," in Proc, IEEE WOWMOM, vol-6, pp. 1–6,2011.

[19]. S. Ruj and B. Roy, "Key Predistributions using Combinatorial Designs for Grid-group Deployment Scheme in Wireless Sensor Networks," ACM Transactions, Sensor Network, vol. 6, no. 4, pp. 1–4:28, Jan. 2010.

[20]. E. F. Assmus and J. D. Key, "Designs and their Codes," Cambridge tracts in mathematics, Cambridge university Press, vol-4, pp.1-6, 1992.

[21]. Vivek Kapoor, Vivek Sonny Abraham and Ramesh Singh, "Elliptic Curve Cryptography", ACM Ubiquity, vol. 0, Issue 20, May 20-26, 2008.

[22]. A. Betten, D. Betten, and V. D. Tonchev, "Unitals and codes,"" Discrete Mathematics, vol. 267, no. 1-3, pp. 23–33, 2003