Blom's Scheme of Pairwise key Distribution Technique to prevent the Sybil Attack in Wireless Sensor Network

Project report submitted in partial fulfilment of the requirement for the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Shantanu Saini-131216

Under the Supervision of

Sh. Amol Vasudeva

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Blom's Scheme of Pairwise key Distribution"** in fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2016 to April 2017 under the supervision of **Amol Vasudeva** (Assistant Professor (Grade-II), Computer Science and Engineering).

The matter embedded in the report has not been submitted for the award of any other degree or diploma.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

( )

Shantanu Saini

Roll No.-131216

( )

Amol Vasudeva

Assistant Professor (Grade-II)

Computer Science and Engineering

Dated:

Acknowledgement

The Blom’s Scheme of Pairwise key Distribution marks the apotheosis of all the concepts comprehended while studying concepts of Security in Wireless Sensor Networks . It has presented us with an opportunity to use the technical know-how to create a system simulation.

Learning through the project under the guidance of our esteemed mentor Sh. Amol Vasudeva, whose expertise knowledge in the domain of Mobile Ad Hoc Networks as well as Sensor Networks, not only cleared all our ambiguities but also generated a high level of interest and gusto in the subject . We are truly grateful for his guidance and support throughout the project. We would also like to thank our Head of the Department, Brig (Retd.) SP Ghrera for is undying faith in the department of computer science and allocation of the project as well as its resources.

The prospect of working in a group with high level of accountability fostered a spirit of teamwork and created a feeling of oneness which thus, expanded our ken, motivated us to perform to the best of our ability and create a report of the highest quality.

To do the best quality work, with utmost sincerity and precision has been our constant endeavor.

Date: Shantanu Saini (131216)

Table of Content

LIST OF ABBREVIATIONS

S.NO.	ABBREVIATION	FULLFORM
1	WSN	Wireless Sensor Networks
2	SSL	Secure Sockets Layer
3	DHL	Distributed Hash Table
4	ASHRAE	American Society of Heating Refrigerating and Air-conditioning Engineers
5	DFD	Data Flow Diagram

List of Figures

LIST OF TABLES

Abstract

In this project, cryptography is implemented using pairwise key generation technique. When a node wants to communicate with some other node in the network without sharing the information with any other node in the network, it will first check whether the node is in its range or not . If they are not in range, they can communicate implicitly i.e. via a node in between them. They both will then find a private key between them. Using this key, they will be able to share the personal information or data. After receiving the message, the key will be destroyed. In cased of a sybill attack a malicious node attack the whole wireless sensor network and Blom's Scheme will help up to detect and destroy any malicious node in the network . In this project, we can take the number of nodes as per our choice.

Chapter-1

INTRODUCTION

1 Introduction

In the recent years there has been an emminent effort to enhance the security in Wireless Sensor Networks in defence & other autonomous organizations.

A Wireless Sensor Network is a collection of sensor nodes which are wirelessly linked. Key establishment is the technique by which two (or more) entities establish a shared secret key. In Blom’s scheme, a trusted party(which is accoiuntable to whole network) gives each participant a secret key and a public identifier, which enables any two participants to independently create a shared key for communicating.

To accomplish security in remote sensor systems, it is essential to have the capacity to encode and confirm messages sent among sensor hubs. Because of asset limitations, accomplishing such key understanding in remote sensor systems is non-trifling. Blom's plan is a symmetric key trade convention utilized as a part of cryptography. However, the worries with Blom's plan are the intricacy required in calculation and also memory use. In this paper, we propose another key pre-dissemination conspire by altering Blom's scheme which decreases the computational multifaceted nature and also memory use.

1.1 PROBLEM STATEMENT

Develop a system which computes a private key between two nodes in order to communicate in a Wireless Sensor Network using Modified Blom's Scheme and detect if any malicious node enters the system and then destroy it. It is made in such a way that no other node will be able to know the computed key.

1.2 OBJECTIVE AND SCOPE OF THE PROJECT

It offers good features on network operation with low resource constrains, data aggregation, and security . This research focuses on key generation in wireless sensor networks and using the key generation technique to detect and destroy any malicious node in the network. The fundamental objective is on the change of security. Different necessities, for example, the adaptability and versatility are additionally considered. The primary undertakings of this examination are as per the following:

1) To analyse the requirements of wireless sensor network key generation and existing problems. The main key generation schemes are assessed on security, efficiency and operation requirements. The features of each scheme are outlined.

2) To propose a modified key generation scheme based on existing schemes. It includes key generation, key usage and key destroy. The proposed scheme should satisfy many requirements of key generation as possible . Using the technique detection of any malicious node in the network and destruction of such node.

3) To analyse and evaluate the proposed key generation scheme on security strength and performance. Simulation results are required to back the analysis.

1.3 ORGANISATION

Chapter 1 highlights the important aspects of the Wireless Sensor Networks which are related to the Blom's Scheme. In this chapter, details of the Sybil attack and a mechanism to achieve security in wireless sensor networks is discussed. The main focus on modified Blom's Scheme is done so that proposed work can be further continued.

Chapter 2 contains the detailed literature overview of the reviewed research papers, books, journals, and conferences. In this chapter, summarized version of the research papers on Blom's scheme of Pairwise Key Distribution Technique to prevent Sybil in Wireless Sensor Networks are presented.

Chapter 3 covers the System Analysis which explains important characteristics of the project with the help of UML diagrams like data flow diagram, class diagram, and sequence diagram

Chapter 4 is the Performance Analysis which contains the screenshots of the developed application. This chapter also lists the resource requirements needed for the application to run.

Chapter 5 closes with the point by point conclusion and extent without bounds work which will be utilized as a managing device for other research researchers to improve the ebb and flow work with higher proficiency.

CHAPTER 2
LITERATURE SURVEY

The implementation of this project requires an extensive knowledge of-

1) Wireless Sensor Networks
2) Sybil Attack
3) Cryptography
4) Blom's Scheme
5) Modified Blom's Scheme

2.1 Wireless Sensor Networks

Wireless sensor networks(**WSN**), aka **wireless sensor and actuator networks** (**WSAN**),are spatially distributed autarchic sensors to *oversee* physical or environmental conditions,such as temperature, sound, pressure, etc.and to collaboratively pass statistics through the network to the main location.

The evolution of wireless sensor networks was motivated by defence applications such as battlefield monitoring; today these networks are used in many industrial and consumer applications ,such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is a collection of "nodes" – ranging from a few to several hundred or even thousand, where each node is inter-connected to other sensors. Each such sensor network node typically consists of three parts : a radio transceiver with an internal antenna or connection to an external antenna , an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting.

2.1.1 Applications of WSN’s

1) Area monitoring

Area monitoring is the most common application of WSNs. In area monitoring, the Wireless Sensor Network is deployed over a stretch of land/region where some activity is to be recorded. Armed forces inspection is an example of use of sensors detects enemy invasion; a civilian example is the geo-fencing of gas or oil pipelines.

2) Health care monitoring

The medical applications of Wireless Sensor Network can be of two types wich includes wearable type and the implanted type . Wearable devices are used on the body surface of a human or just at near the user. The implantable medical devices are the devices which are embedded into the human body. There are many other applications of the WSN e.g. body position measurement and determining the location of the person, overall monitoring of ill and unfit patients in hospitals , penitentiaries and homes. Body area networks can collect information about an individual's health, fitness, and energy expenditure.

3) Air pollution monitoring

Wireless sensor networks have been used in a number of sites to monitor the concentration(ppm) of dangerous gasses and substances for citizens. Ad hoc wireless links have an advantage rather than wired installations, which make them more mobile and provide ease for testing readings in different areas.

4) Forest fire detection

A network of Sensor Nodes installed in a forest can be used to detect whether a fire has started. The nodes equipped with sensors can be used to measure temperature-change, humidity, and concentration and nature of gasses which are being produced during a forest-fire. Thanks to Wireless Sensor Networks, the early detection is very important firefighters to take any action, the fire brigade will be able to know about the forest fire how it is generated and what is the area affected.

5) Landslide detection

A landslide detection system can make use of a WSN to detect the movements of soil-form and changes in various environmental parameters that may occur before or during a landslide. The data gathered using the WSN can be used to detect the occurance of landslide before happening.

6) Water quality monitoring

Water quality monitoring involves analyzing properties water properties in various water bodies. Use of many wirelessly distributed sensors enables the creation of a more accurate map of the water status, and allows the permanent deployment of monitoring stations in locations with difficult access, without the manual retrieval of data.

7) Natural disaster prevention

Wireless sensor networks can be used to prevent many of the natural disasters, like drought , floods , femine and earthquake by predetermining the data from the environment and using it to monitor association of one activity with other.

8) Machine health monitoring

WSN have been developed for machinery and condition-based maintenance (CBM) .They offer significant cost reduction and enable new functionalities to the system.

Wireless sensors can be placed in remote locations which are difficult or impossible to reach with a wired system.

9) Data center monitoring

Due to the high number of servers in a data center, cabling and IP Address generation is a big issue . To overcome that problem of wiring , more and more servers are fitted out with wireless sensors to sense the temperature, so as to monitor the temperatures of racks of servers. As per ASHRAE recommendation , up to 6 temperature sensors per rack of server, gives an advantage compared to traditional cable sensors.

10) Data logging

WSN are also used for the data collection and monitoring of environmental information, this can be as simple as the monitoring of the temperature in a room to the level of water in tanks in power plants. The statistical information can then be used to show how systems have been working. The main advantage of WSNs over conventional system is the live data that is being recorded.

11) Water/wastewater monitoring

Monitoring the quality and level of water includes many activities such as checking the quality of underground or surface water and ensuring a country's water infrastructure for the benefit of both human and animal. It may be used to protect the wastage of water.

.

2.2 Sybil Attack

The Sybil Attack is an attack in which the framework is influenced by making various characters in Wireless Sensor Network . It is named after the subject of the book Sybil , which is a contextual investigation of a lady determined to have dissociative personality issue. The name was recommended in or before 2002 by Brian Zill at Microsoft Research.

In a Sybil attack, the aggressor enters the arrangement of a WSN by making an extensive number of pseudo-unknown personalities, utilizing them to pick up an excessively expansive impact. A framework's weakness to being assaulted by a Sybil assault relies on upon the method utilizing which personalities can be produced, how much the framework acknowledges contributions from substances that don't have a chain of trust connecting them to a trusted element, and whether the notoriety framework treats all elements indistinguishable.

A substance on a Peer to Peer system is a bit of intuitive programming having admittance to neighborhood assets. An element is made accessible to all different substances of the system utilizing a specific personality. More than one character can compare to a solitary substance. As such, the mapping of personalities to the substances in the WSN is many to one.

A broken hub or an enemy may introduce different personalities to a distributed system with a specific end goal to show up and work as numerous particular hubs. In the wake of winding up noticeably some portion of the distributed system, the enemy may then catch correspondences or act vindictively. By disguising and displaying numerous personalities, the enemy might have the capacity to influence voting results or even significantly control the system.

With regards to (human) online groups, such various characters are here and there known as sockpuppets.

2.2.1 Example of Sybil Attack

A notable Sybil attack (in conjunction with a traffic confirmation attack) was launched against the Tor anonymity network for several months in 2014 by unknown perpetrators. Many in the network security community suspect the NSA/CIA to be responsible for the attack, and some speculate that the attack may have been connected to the investigation into the Silk Road website.

2.2.2 Prevention of Sybil Attack

Validation techniques can be utilized to counteract Sybil assaults and reject disguising unfriendly elements. A nearby element may acknowledge a remote character in view of a focal expert which guarantees a balanced correspondence between a personality and a substance and may even give a switch query. A personality might be approved either straightforwardly or by implication. Aberrant approval, the nearby substance inquiries the focal specialist to approve the remote characters. In aberrant approval, the neighborhood element depends on officially acknowledged personalities which thusly vouch for the legitimacy of the remote character being referred to.

Identity based approval procedures, for the most part, give responsibility to the detriment of secrecy, which can be an undesirable exchange off particularly in online gatherings that desire to allow control free data trade and open discourse of delicate themes. An approval expert can endeavor to protect clients' secrecy by declining to perform turn around queries, yet this approach makes the approval specialist a prime focus for assault. On the other hand, the specialist can utilize some system other than information of a client's genuine character -, for example, check of an unidentified individual's physical nearness at a specific place and time - to implement a coordinated correspondence between online personalities and true clients.

2.3 Cryptography

Crypto is gotten from the Greek word "kryptós" which signifies "covered up" or "mystery", and graphy is gotten from the Greek word "graphein" which signifies "composing". Cryptography is the practice and investigation of procedures for secure correspondence within the sight of outsiders called foes. All the more by and large, cryptography is about developing and investigating conventions that avoid outsiders or people in general from perusing private messages; different parts of data security, for example, information secrecy, information uprightness, and verification, are fundamental to present day cryptography. Present day cryptography exists at the crossing point of the orders of arithmetic ,software engineering, and electrical building. Uses of cryptography incorporate ATM card

Cryptography before the modern age was adequately synonymous with encryption, the change of data from a decipherable state to evident drivel. Current cryptography is intensely in view of scientific hypothesis and software engineering hone; cryptographic calculations are planned around computational hardness presumptions, making such calculations difficult to soften up practice by any foe. It is hypothetically conceivable to break such a framework, yet it is infeasible to do as such by any known functional means.

2.3.1 Why Cryptography is necessary for a Distributed System

Supporting the offices of a circulated framework, for example, asset dissemination, requires the utilization of a hidden message passing framework. Such frameworks are, thus, dependent on the utilization of a physical transmission arrange, whereupon the messages may physically be imparted between hosts.

Physical systems and, consequently, the fundamental message ignoring frameworks assembled are defenseless against attack. For instance, hosts may effectively connect to the system and tune in on the messages (or 'discussions') being held. On the off chance that the transmissions are in a promptly justifiable shape, the busybodies might have the capacity to choose units of data, in actuality taking their data content.

Besides the robbery of user’s information, which might be in itself of extraordinary esteem, there may likewise be framework data being passed around as messages. Meddlers from both inside and outside the framework may endeavor to take this framework data as methods for either breaking interior get to limitations or to help in the assault of different parts of the framework. Two perhaps more awful situations may exist where the assaulting framework may alter or embed fake transmissions on the system. Tolerating faked or changed messages as substantial could lead a framework into disarray.

Without sufficient security procedures, Distributed Systems are amazingly helpless against the standard sorts of assault illustrated previously. The encryption strategies talked about underneath mean to give the missing assurance by changing a message into a frame where on the off chance that it was captured in travel, the substance of the first message couldn't be expressly found. Such encoded messages, when they achieve their expected beneficiaries, nonetheless, are equipped for being changed once more into the first message.

2.3.2 Types of key encryption

1. Public Key Encryption

Asymmetric encryption, also known as public-key encryption utilizes a pair of keys – a public key and a private key. On the off chance that you scramble information with people in general key, just the holder of the relating private key can decode the information, henceforth guaranteeing classification.

Many "secure" online exchange frameworks depend on asymmetric encryption to build up a safe channel. SSL, for instance, is a convention that uses asymmetric encryption to give correspondence security on the Internet.

Asymmetric encryption calculations normally include exponential operations, they are not lightweight as far as execution. Hence, topsy-turvy calculations are frequently used to secure key trades as opposed to utilized for mass information encryption.

2. Private Key Encryption

Symmetric encryption, as the name recommends, implies that the encryption and unscrambling operations use a similar key. For two imparting parties utilizing symmetric encryption for secure correspondence, the key speaks to a common mystery between the two.

2.4 Blom’s Scheme

Blom's scheme is a symmetric threshold key exchange cryptography protocol. It permits any pair of clients in the framework to locate a one of a kind shared key for secure correspondence. In this plan, a system with N users and a crash a fewer than t+1 users can't uncover the keys which are held by different users. Consequently, the security of the system relies on upon the picked estimation of t, which is called Blom's protected parameter (t<<N). A Larger estimation of t prompts more noteworthy versatility however a high esteem builds the measure of memory required to store key data.

Generation of the Public grid: Initially, a focal expert or base station initially develops a (t + 1) ×N framework P over a limited field GF (q), where N is the extent of the system and q is the prime number. P is known to all clients and it can be built utilizing a Vandermonde lattice. It can be demonstrated that any t+1 sections of P are straightly autonomous when i=1, 2,… N are all particular.

$$P = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ n_1 & n_2 & n_3 & \dots & n_N \\ {n_1}^2 & {n_2}^2 & {n_3}^2 & \dots & {n_N}^2 \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ {n_1}^t & {n_2}^t & {n_3}^t & \dots & {n_N}^t \end{bmatrix}$$

Figure 1: Vandermode matrix

of Secret Key (Private grid): The focal expert or the base station chooses an arbitrary (t + 1) × (t + 1) symmetric lattice S over GF (q), where S is mystery and just known by the focal specialist. A × (t + 1) framework A = (S. P)T is processed which is required for producing the common key. K =A. P = (S. P) T. P = PT. ST.P = PT.S.P = (A.P)T = KT.

Generation of shared key by the user pair: User pair (i, j) will use K_{ij}, the element in row I and column j in K, as the shared key. Because K_{ij} is calculated by the ith row of A and the jth column of P, the central authority assigns the ith row of A matrix and the ith column of P matrix to each user i, for 1, 2….. N. Therefore, when user i and user j need to establish a shared key between them, they first exchange their columns of P, and then they can compute K_{ij} and K_{ji}, respectively, using their private rows of A. It has been proved in that the above scheme is t-secure if any t + 1 columns of G are linearly independent. The t-secure parameter guarantees that no compromise of up to t nodes has any information about K_{ij} or K_{ji}.

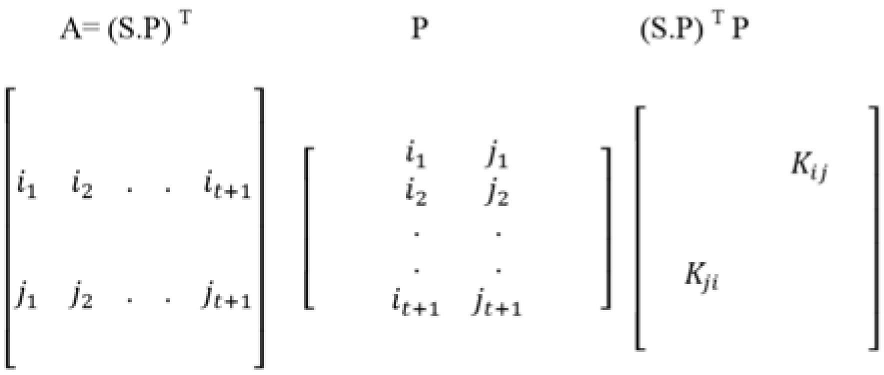


Figure 2: Generating keys in Blom’s Scheme

2.4.1 Example of Blom's Scheme

The example below shows the working of proposed Scheme that uses a random matrix as a public key and generates a private key. Let us consider a network with 4 nodes and the following parameters:

1. Let C be the Central Authority or Base Station.
2. Secure parameter t = 3, which says if more than 3 nodes in the network are compromised, it is not possible to find the keys of other users.
3. Prime number q= 31.
4. As the public matrix (P) should be of the order (t+1) * (t+1), P can be taken as any random (3+1) * (3+1) matrix.

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 3 & 1 \\ 4 & 0 & 9 & 5 \end{bmatrix}$$

5. The secret symmetric matrix can be obtained as follows:
   Let us take a random matrix,

$$A=\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 2 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 3 & 1 \end{bmatrix} \qquad B=\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 2 & 0 & 2 \\ 1 & 0 & 1 & 3 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

We obtain the secret matrix S by taking the product of two matrices A and B,

$$S=\begin{bmatrix}3 & 2 & 2 & 4\\ 2 & 6 & 1 & 5\\ 2 & 1 & 2 & 4\\ 4 & 5 & 4 & 14\end{bmatrix}$$

6. Now, we calculate matrix A using,
   $A = (S.P)^T \bmod q$

$$(S.P)=\begin{bmatrix}25 & 8 & 53 & 36\\ 30 & 5 & 60 & 40\\ 23 & 6 & 49 & 31\\ 73 & 12 & 155 & 95\end{bmatrix}$$

$$A=(S.P)^T \bmod 31=\begin{bmatrix}25 & 8 & 22 & 5\\ 30 & 5 & 29 & 9\\ 23 & 6 & 18 & 0\\ 11 & 12 & 0 & 2\end{bmatrix}$$

7. Once A is calculated, each sensor node memory is filled with unique row chosen from A with the corresponding index. These are the private keys for the nodes.

$K_{A,B}$ =

$$\begin{bmatrix}8 & 5 & 6 & 12\end{bmatrix}\begin{bmatrix}3\\ 1\\ 3\\ 9\end{bmatrix}$$

$K_{B,A}$ =

$$\begin{bmatrix}22 & 29 & 18 & 0\end{bmatrix}\begin{bmatrix}2\\ 0\\ 1\\ 0\end{bmatrix}$$

As seen above, both the nodes generate a common key and further communication can be done using the shared key generated.

2.4.2 Disadvantages of Blom's Scheme

(1) Any two sensors can set up a pairwise key when there are no traded off sensors

(2) Even with some nodes compromised, the others in the network can still establish pairwise keys;

(3) A node can see the usual keys to specify whether or not it can establish a pairwise key and thereby help reduce communication overhead.

2.5 Modified Blom's Scheme

There are two ways to implement Modified version of the Blom's Scheme

Method 1:

In this technique, we make utilization of the Bloms plot. As expressed before the Blom's plan makes utilization of Vandermonde grid which is an open framework and this network is in charge of all calculations in producing keys and since it's an open lattice, it could be known even to the meddlers. We pick every one of the qualities or components of this network to be particular so that any $\lambda + 1$ sections of G are directly autonomous i.e., to produce extraordinary keys. However, when the estimation of λ increments to bigger qualities, then the quantity of lines of the general population framework increment and this, thus, brings about a more noteworthy incentive in the sections in light of the fact that the segment values increment in a geometric arrangement. We know in remote sensor systems, sensor hubs contains restricted memory and vitality, wherein Blom's plan for any two hubs to create a typical key, every hub ought to store section of open grid and line of the computed mystery framework and this would be troublesome for sensor systems to store both the line and segment in the memory for a substantial system.

To reduce the computation and memory overhead in Blom's scheme, instead of using Vandermonde matrix we propose the use an Adjacency Matrix as the public matrix. As the Adjacency matrix is a square matrix with 1s and -0s, it reduces the complexity of calculating values for all the elements corresponding to the columns in Vandermonde matrix. This adjacency matrix is formed in such a way that all nodes that are neighbors of a particular node are filled with 1s and remaining with q-1(since public matrix cannot contain 0s).

Another advantage of using Adjacency matrix is it reduces the cost of saving the columns in the memory of sensor because any node can easily generate Adjacency matrix of known size. Similar to Blom’s scheme the operation which are to be performed to generate the keys will depend on the prime number i.e., the number which depends on the desired key length.

$$\begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{vmatrix}$$

Figure 3: Adjacency Matrix

$$\begin{vmatrix} 1 & 1 & 28 & 1 \\ 1 & 28 & 28 & 28 \\ 28 & 28 & 1 & 1 \\ 1 & 28 & 1 & 28 \end{vmatrix}$$

Figure 4: Changed modified Adjacency Matrix for Modulo 29

The first parallel shape Adjacency Matrix is portrayed in Figure 3. The main change that is made to the Adjacency Matrix is every one of the zeros are supplanted with the q-1(prime number - 1). We can see from Figure 4 that the Adjacency Matrix comprises of just two distinct numbers one's and q-1(prime number - 1), thus all the further estimations will be exceptionally straightforward. Key era strategy is like that utilized as a part of the Blom's plan [7]. Taking after are the means required in figuring the key.

Step 1: Generating a G matrix. We first select a primitive element from a finite field GF(q), where q is larger than the desired key length (and also q > N), and then construct adjacency matrix G of size N ×N as discussed in the previous section depending on the node neighbours. Then depending on the λ value first λ rows along with N columns are selected as the public matrix. Let G (j) represent the jth column of G, in Blom’s scheme G (j) is provided to node j but in our case, each node knows its neighbors and therefore the memory usage for storing G (j) at a node is not required.

Step 2: Generating key spaces. The Central Authority generates ω random, symmetric matrices D1, . . . , Dω of size (λ+1)×(λ+1). We then compute the matrix Ai= (Di. G) T. Let Ai(j) represent the jth row of Ai.

Step 3: Computing Secret Key. The central authority stores each row of the matrix A in the node memory with the corresponding index

Method 2:

Blom's scheme is a capable key organization plot yet its inadequacies consolidate costly count overhead and memory cost. We propose another arrangement in this wander changes Blom's arrangement in a way that lessens memory and computation costs.

Let p be a large prime number which is less than the number of nodes in the network. Each n user U in the network is assigned a distinct prime number Ru (mod p). Then the main authority chooses three random numbers a(mod p), b(mod p) and c(mod p). For each user U, the main authority calculates the numbers $A_U = a+b.r_U$ (mod p) and $b_U = b+c.r_U$ (mod p). Each user U forms the linear polynomial $g_U(x) = Au + b_U .x$(mod p). Now, if A wants to communicate with B, then A & B computes $K_{AB} = G_A (R_B)$ and $K_{BA} = G_B (R_A)$ respectively.

Let us understand this with an example

- Consider a network consisting of two users A & B.
- Let p=23 , $r_A = 11$, $r_B = 3$.
- The main authority chooses three random numbers a=8, b=3, and c=1.
- Then, $a_A = 8 + 11*3$(mod 23) = 18 and $b_A = 3 + 11*1$(mod 23) = 14;
- $a_B = 8 + 3*3$(mod 23)=17 and $b_B = 3 + 1*3$(mod 23) = 6.
- Hence, $g_A(x) = 18 + 14x$ and $g_B(x) = 17 + 6x$.
- Now, $K_{AB} = g_A(r_B)$mod 8 = 14 and $K_{BA} = g_B(r_A)$mod 8 = 14.
- Therefore $K_{AB} = K_{BA}$. K_{AB} is the private key between A and B.

CHAPTER-3

SYSTEM ANALYSIS

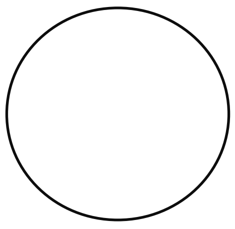
In this section discussed data flow diagram, Entity relationship diagram. These things are represented as diagrams with proper notation.

3.1 Data Flow Diagram

The information stream outline is a standout amongst the most imperative instruments utilized by the framework examiner DeMarco (1978) Nad Gane Sarson (1979) advances the utilization if the information stream graph as displaying apparatuses through their organized framework investigation procedures. An information stream outline ought to be the primary instrument utilized by framework expert to model framework segments..

Processes

Processes demonstrate what framework does. Each procedure has at least one information sources of info and produces at least one information yield, Circles in an information stream outline speak to forms. Each procedure has one of a kind name and number. This name and number show up inside the circle that speaks to the procedures in an information stream outline. This procedure is spoken to as a circle.



Data Stores

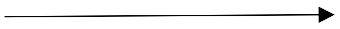
Document or information store is the archive of information. They contain information that is held in the framework. Procedures can enter the information store or recover information from the information store.

External Entities

Outer elements are outside the framework yet they either supply input information into the framework or utilize the framework yield, they are substances which the planner has no control. Square or rectangle may speak to outer substances that supply information into a framework or now and then called sources.

Data Flows

Information stream, show, the section of information in the framework and are spoken to lines joining framework parts. A bolt demonstrates the heading of the stream and the line named by the name of the information stream.



3.1.1 DFD LEVEL 0

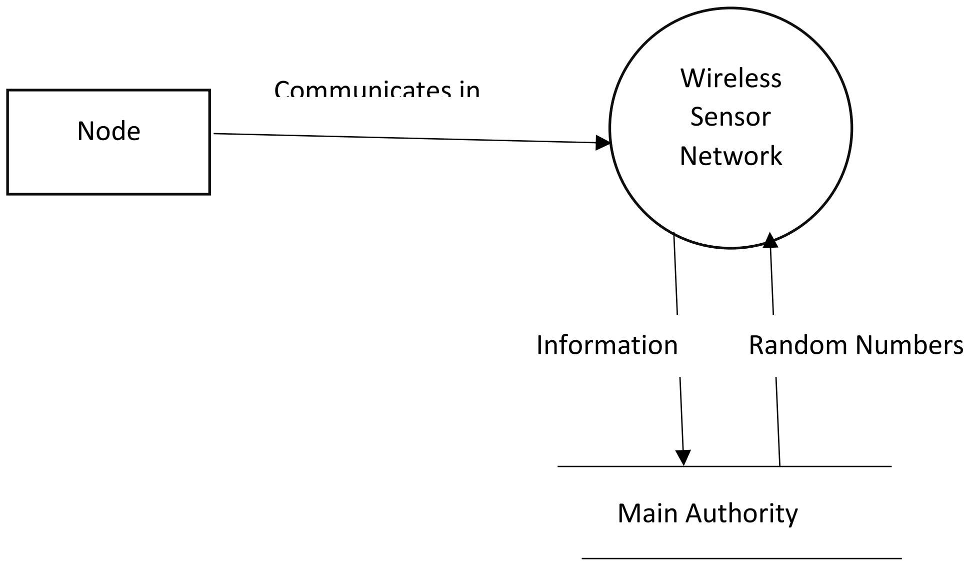


Figure 3: DFD Level 0

3.1.2 DFD LEVEL 1

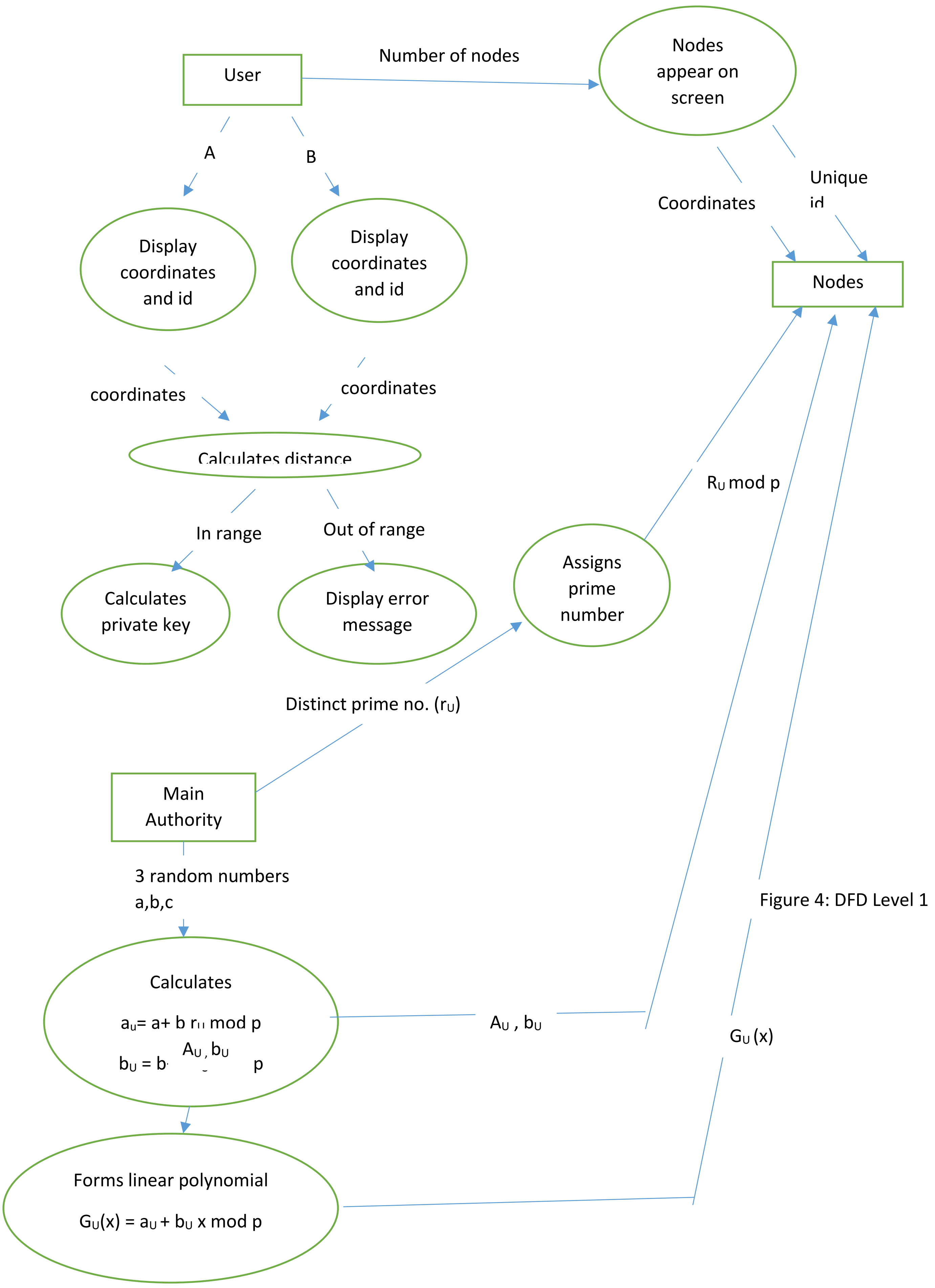


Figure 4: DFD Level 1

3.2 Class Diagram

Class Diagram are visual portrayals of the static structure and creation of a specific framework utilizing the traditions set by the Unified Modeling Language (UML). Out of all the UML chart sorts, it is a standout amongst the most utilized ones. Framework architects utilize class outlines as a method for disentangling how questions in a framework cooperate with each other. Utilizing class graphs, it is simpler to portray every one of the classes, bundles, and interfaces that constitute a framework and how these segments are interrelated.

Classes in class diagrams are represented by boxes that are partitioned into three:

1. The top partition contains the name of the class.
2. The middle part contains the class’s attributes.
3. The bottom partition shows the possible operations that are associated with the class.

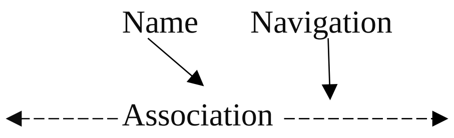
Relationships in Class Diagrams

Classes are interrelated to each other in specific ways. In particular, relationships in class diagrams include different types of logical connections. The following are such types of logical connections that are possible in UML:

- Association
- Directed Association
- Reflexive Association
- Multiplicity
- Aggregation
- Composition
- Inheritance/Generalization
- Realization

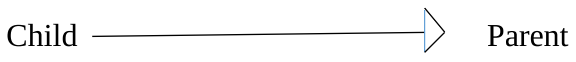
Association

Association is spoken to by a spotted line with (without) bolts on both sides. The two finishes speak to two related components as demonstrated as follows. The assortment is likewise said at the finishes (1, * and so forth) to show what number of items are related.



Generalization

Generalization describes the inheritance relationship of the object oriented world. It is parent and child relationship.It is parent and tyke relationship. Speculation is spoken to by a bolt with empty sharpened stone as demonstrated as follows. One end speaks to the parent component and the flip side tyke component.



Composition

The graphical portrayal of a creation relationship demonstrates organization as a diamond shape on the containing class end of the lines that associate contained classes to the containing class.

Class Diagram

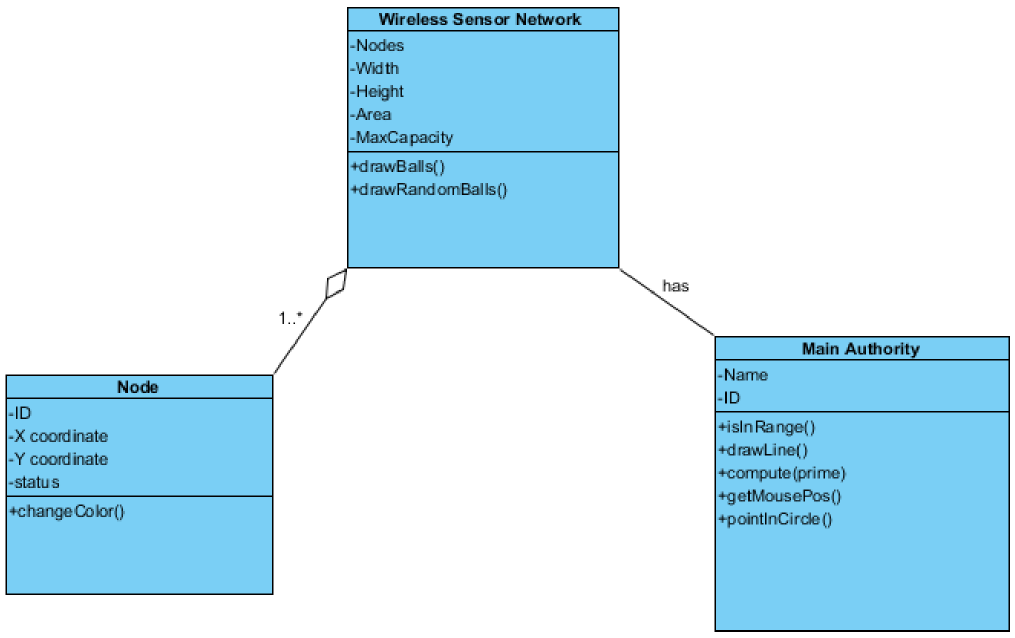


Figure 5: Class Diagram

3.3 Use Case Diagram

The fundamental motivation behind an utilization case outline is to show who communicates with your framework, and the principle objectives they accomplish with it.

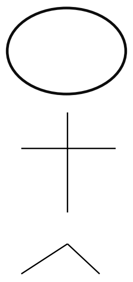
Use case diagrams consist of 4 objects.

- Actor
- Use case
- System
- Package

The objects are further explained below.

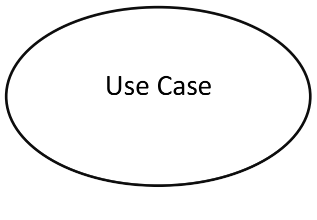
Actor

Actor in a use case diagram is any entity that performs a role in one given system. This could be a person, organization or an external system and usually drawn like skeleton shown below.



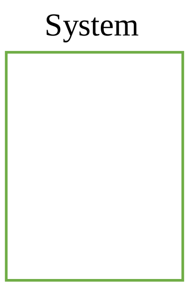
Use Case

A usezcase **representsxa function or an action within the system**. It's drawn as an oval and named with the function.



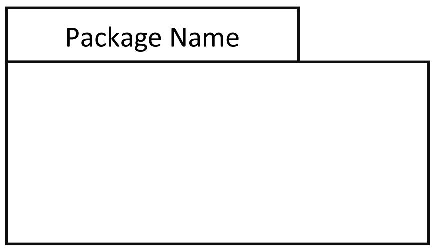
System

The system is utilized to characterize the extent of the utilization case and drawn as a rectangle. This a discretionary component however valuable when you're picturing vast frameworks. For instance, you can make all the utilization cases and afterward utilize the framework protest characterize the degree secured by your venture.



Package

The package is another discretionary component that is greatly valuable in complex charts. Like class charts, bundles are utilized to assemble together utilize cases. They are drawn like the picture demonstrated as follows.



Use Case Diagram

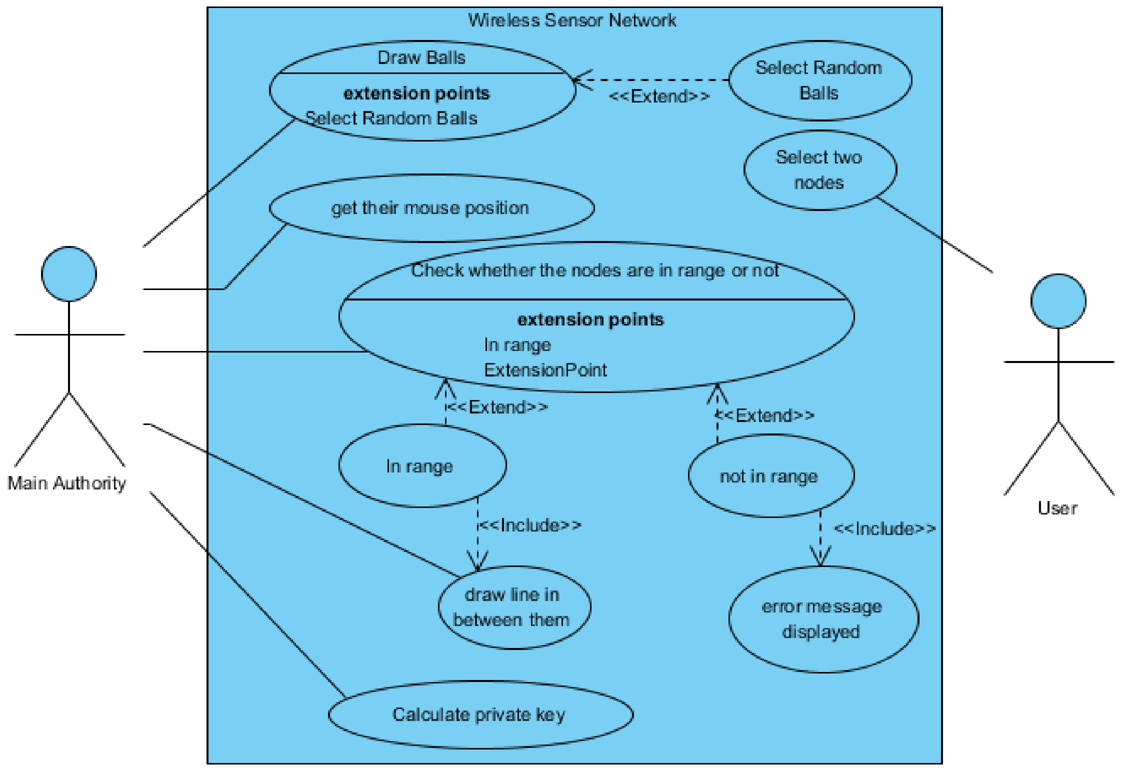


Figure 6: Use Case Diagram

3.4 Sequence diagram

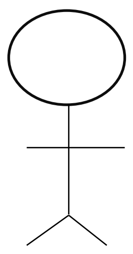
A Sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence.

It consists of following objects-

- Actor
- Messages
- Lifelines
- Loops

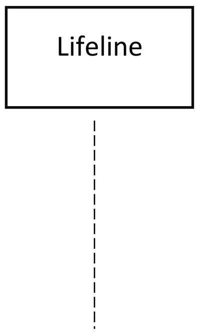
Actor

An Actor models a sort of pretended by an element that communicates with the subject (e.g., by trading signs and information), however which is outer to the subject (i.e., as in a case of a performer is not a piece of the case of its relating subject). On-screen characters may speak to parts played by human clients, outside equipment, or different subjects. Take note of that a performing artist does not really speak to a particular physical substance but rather just a specific feature (i.e., "part") of some element that is important to the determination of its related utilize cases. Along these lines, a solitary physical example may assume the part of a few unique performing artists and, alternately, a given on-screen character might be played by various distinctive occasions.



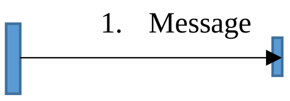
Lifelines

Lifelines are vertical dashed lines that indicate the object's presence over time.



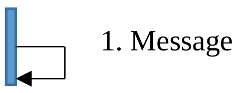
Messages

Messages are bolts that speak to correspondence between articles. Utilize half-arrowed lines to speak to Asynchronous messages. Asynchronous messages are sent from a protest that won't sit tight for a reaction from the recipient before proceeding with its assignments.



Loops

A redundancy or loop inside a diagram is delineated as a rectangle.



Sequence Diagram

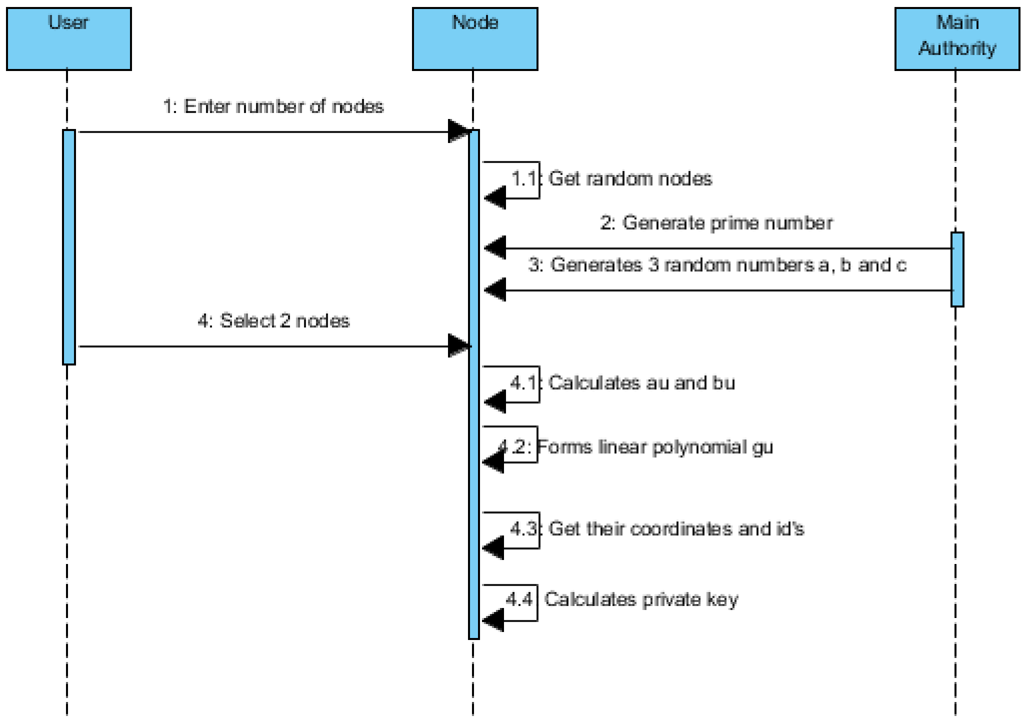

User
Node
Main Authority
1: Enter number of nodes
1.1: Get random nodes
2: Generate prime number
3: Generates 3 random numbers a, b and c
4: Select 2 nodes
4.1: Calculates au and bu
4.2: Forms linear polynomial gu
4.3: Get their coordinates and id's
4.4: Calculates private key

Figure 7: Sequence Diagram

3.5 ALGORITHM

1. Each of n users is given initial secret keying material and public data.

2. Each pair of users UA, U_B may compute the secret key K_{AB}=K_{BA}

3. Let p be a large p≤n. everyone has knowledge of the prime p.

4. Each n user U in the network is assigned a distinct public number r_{un}(mod p).

5. Main authority chooses three secret random numbers a, b, and c(mod p).

6. For each user, U. Trent calculates the numbers

$$a_u = a+b.r_u(\text{mod } p) \qquad b_u = b+c.r_u(\text{mod } p)$$

And sends them via a secure channel

7. Each user U forms the linear polynomial

$$g_u(x) = a_u + b_u.x(\text{mod } p).$$

8. If A wants to communicate with B, then A computes

$K_{AB} = g_A(r_B)$, while B computes $K_{AB} = g_B(r_A)$, which is the private encryption key.

3.6 SIMULATION ENVIRONMENT

Summary of the Simulation Environment	
Simulator	JavaScript
Simulation Area	640 pixels * 480 pixels
Allocation	Dynamic
Transmission Range of Legitimate node	200 pixels
Number of nodes	2 nodes – 80 nodes
An observation period	Variable, by selecting different number of nodes
Communication Direction	0-2π
Number of nodes communicating simultaneously	2
Presentation of nodes	Simultaneous
Framework	HTML Canvas(p5.js)

Table 1: Simulation Environment

3.7 SYSTEM TESTING

System testing includes the testing of the computational model of the remote sensor hubs delineating the correspondence utilizing Blom's plan and effective running of the created proposed framework. The client tests the created framework and the progressions are made by the necessities of the framework. The testing stage includes the testing of the created framework utilizing different hubs. An intricate testing of information is readied and the framework is tried utilizing an alternate number of hubs. While testing, mistakes are noted and the redresses are made. The rectifications are likewise noted for the future references. The principle expert keeps up the correspondence between the hubs and the hubs are conveyed on the framework.

TESTING

System testing is the phase of usage that is gone for guaranteeing that the framework works precisely and proficiently before the live operation initiates. Testing is imperative to the accomplishment of the framework. Framework testing makes the consistent suspicion that if every one of the parts of the framework are right, then the objective will be effectively accomplished. A progression of testing are accomplished for the proposed framework before the framework is prepared for the client acknowledgment testing.

The following are the types of Testing:

1. Unit Testing
2. Integration Testing
3. Validation Testing
4. Verification Testing
5. User acceptance testing

UNIT TESTING

The procedure level testing is made first. By giving improper inputs, the errors occurred are noted and eliminated. Then the testing is done for the individual node. For example deployment of the nodes in the given region and partial non-overlapping of nodes in a 2-D space. Then the testing is done for the randomly generated nodes. For example, the randomly generated nodes are not overlapping and uniform distribution of the random nodes in the given space. After this unit testing is done for the click event on the given node. For example, each time a click is made the current coordinates of the mouse is computed with each of the nodes for the distance and the distance between the nodes and the current position is compared with the radius for the click detection. After this, the unit testing for the color change event is made when the node is clicked.

INTEGRATION TESTING

Testing is accomplished by every module. In the wake of testing every one of the modules, the modules are incorporated and testing of the last framework is finished with the test information, uncommonly intended to demonstrate that framework will work effectively in every one of its perspectives conditions. Accordingly, the framework testing is an affirmation that all is right and a chance to demonstrate the client that the framework works.

VERIFICATION TESTING

The last stride includes Validation testing, which decides if programming capacities are as client anticipated. The end-client than the framework engineer directs this test. Most programming engineers as a procedure called "Alpha and Beta Testing" to reveal that lone the end client appears to be ready to discover.

VALIDATION TESTING

Validation is a basic idea in programming plan. This is the extension between client prerequisites and a usage that fulfills those necessities.

This is evident in the event that it can be shown that the testing will bring about a usage that shows up a couple of months after the fact. This will make two issues

- The time delay between the cause and appearance of the problem.
- The effect of the system errors on files and records within the system.

USER ACCEPTANCE TESTING

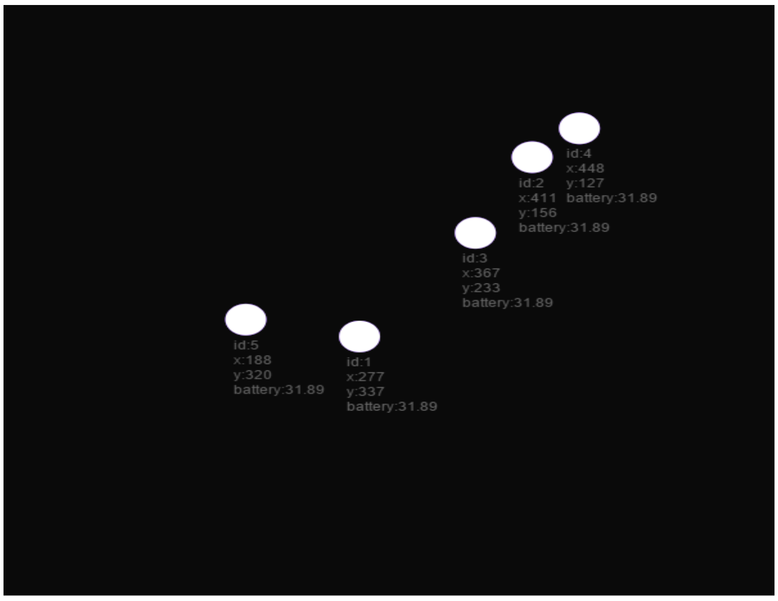
User acknowledgment testing of a framework is the key factor in considering the achievement of the framework. The framework under review is tried for the user acknowledgment by always staying in contact with the forthcoming framework user whenever creating and rolling out improvements at whatever point required.

CHAPTER-4

PERFORMANCE ANALYSIS

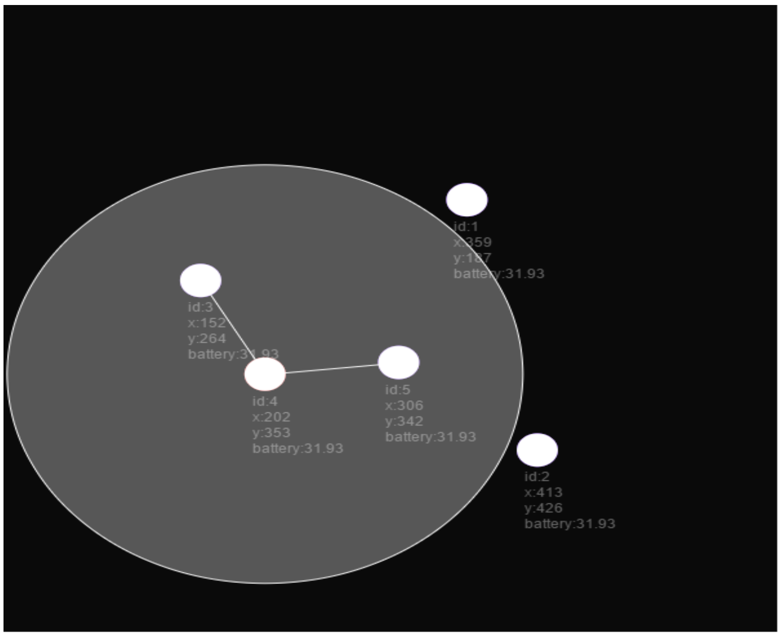
4.1 SCREENSHOTS

1. Run index.html

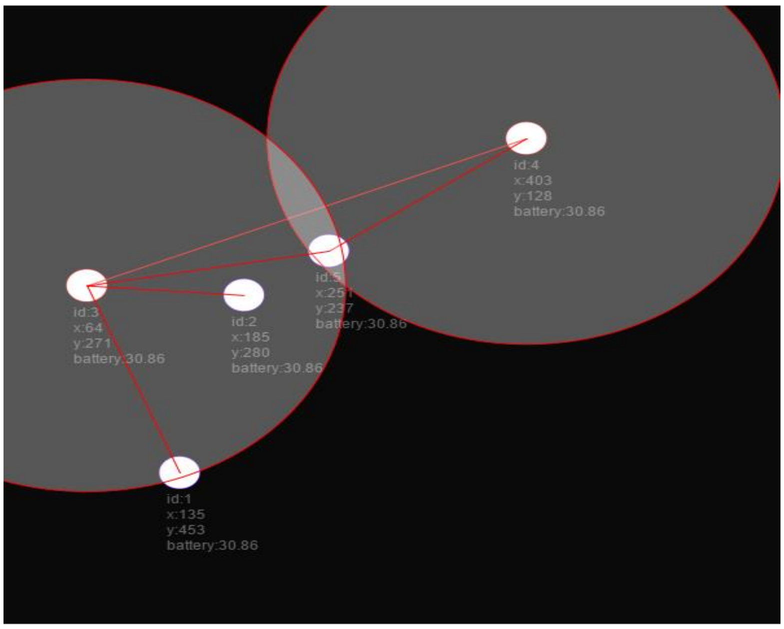


Case 1: When nodes are not in range

1. Select the first node

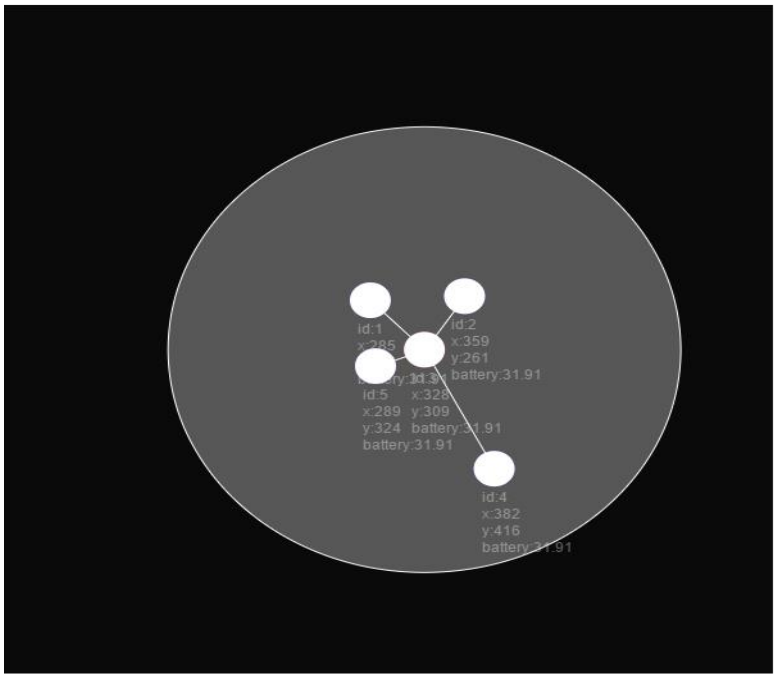


2. Select the second node

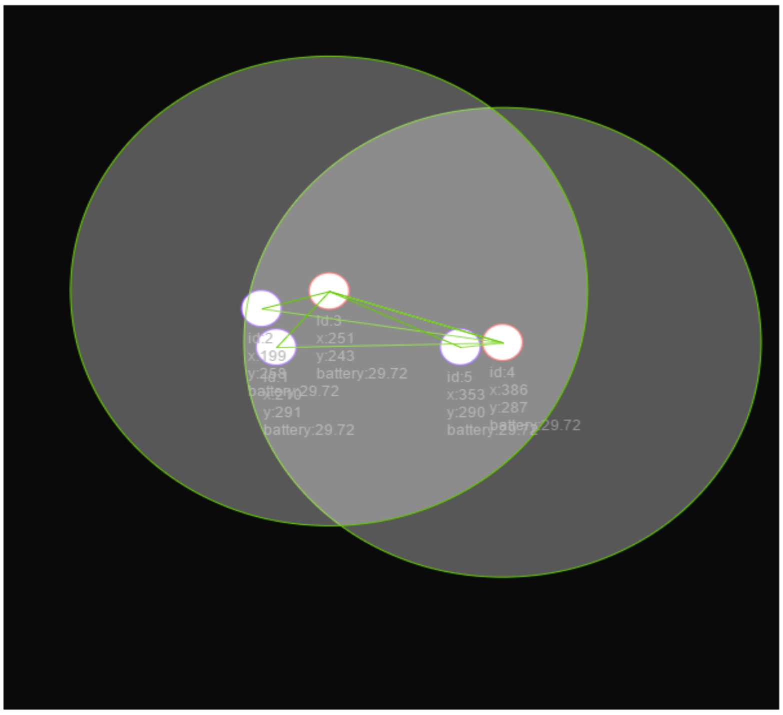


Case 2: When nodes are in range

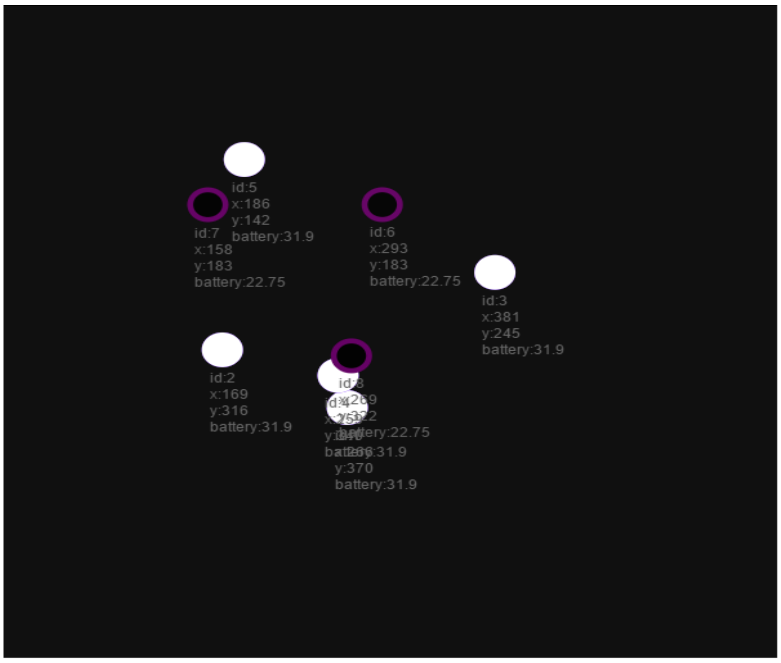
1. Select the first node



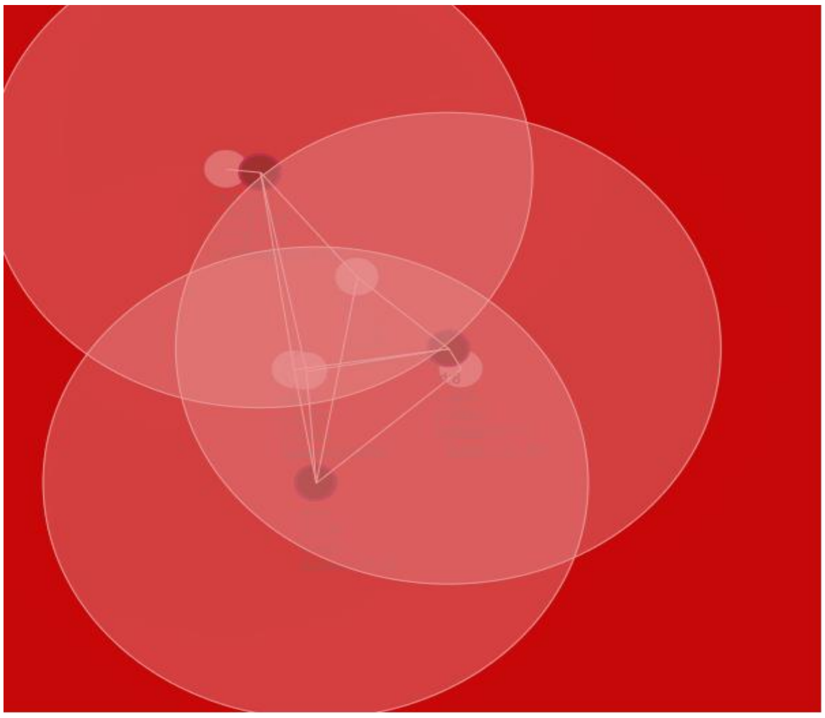
2. Select the second node



2.Intoduce malicious node on network

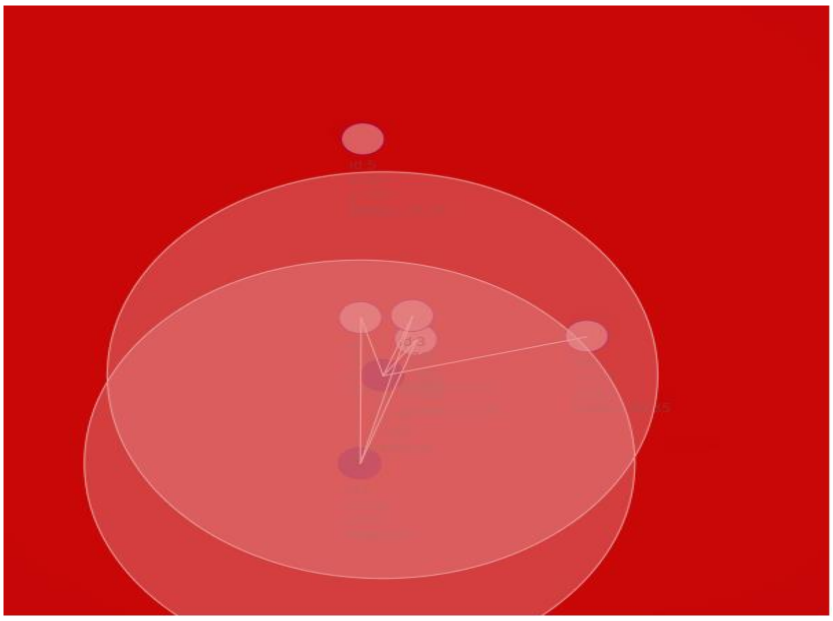


2.1 Malicious node try to establish link with other neighbouring nodes & in return legitimate node identifies malicious node using bloms's scheme

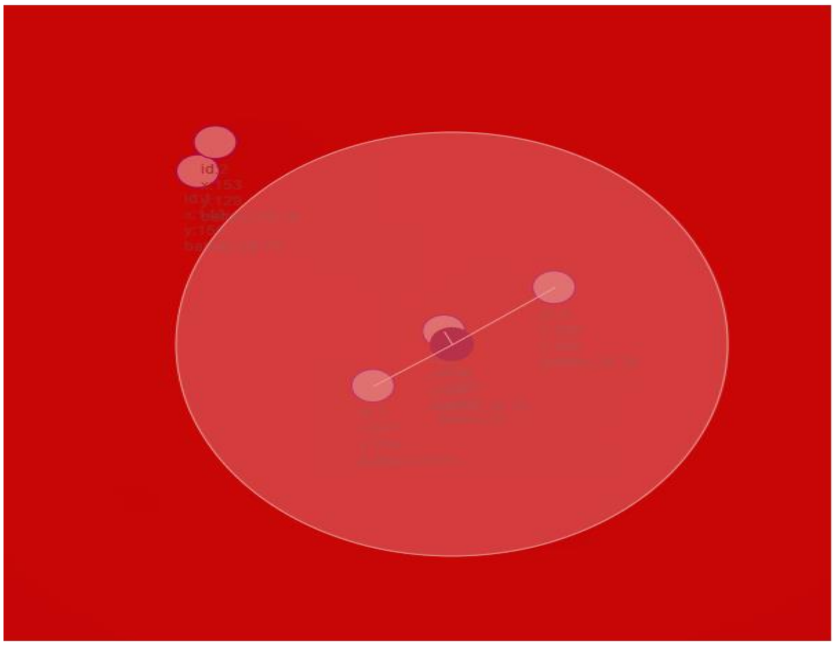


2.2 Legitimate node tries to destroy malicious nodes using the signal strength

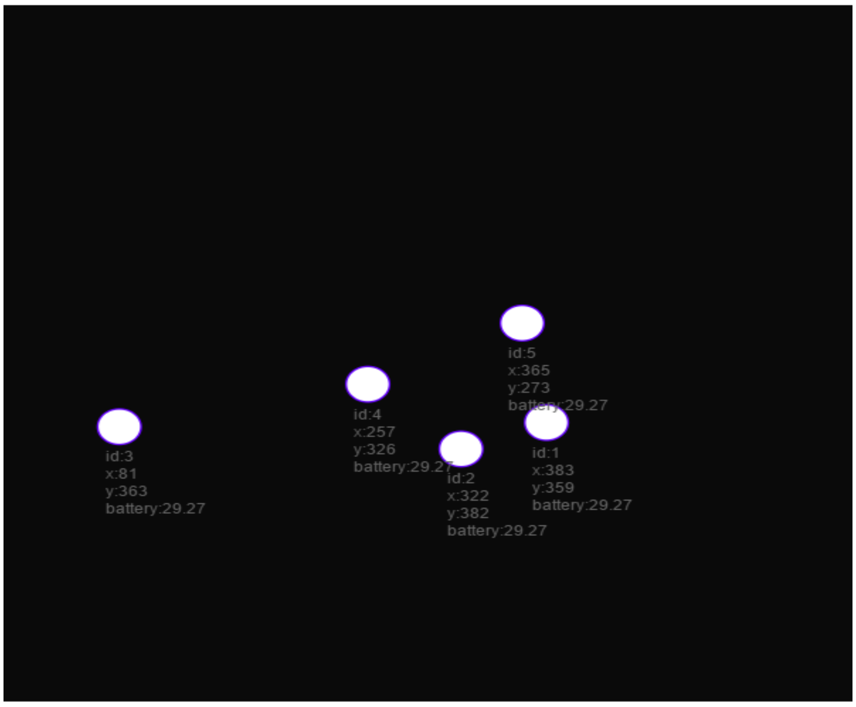
1st malicious node destroyed



2nd malicious node gets destroyed



3rd Malicious node gets destroyed and legitimate nodes broadcast system is secure



4.2 RESOURCE REQUIREMENTS

Hardware Requirements

Processor-Used	PENTIUM III 866Mhz
RAM	256 MB SD RAM
Monitor	15” COLOR (RESOLUTION>640*480)
Hard Disk	20GB
Floppy Drive	NA
CD Drive	NA
Keyboard	STANDARD 102 KEYS
Mouse	2 BUTTONED

Table 2: Hardware Requirements

Software Requirements

Operating System	WINDOWS XP
Environment	.NET Framework 3.5
Language & Web Technology	JAVASCRIPT

Table 3: Software Requirements

CHAPTER-5

Conclusion and Future Scope

It is concluded that the application functions admirably and fulfill the reproduction imperatives. The application is tried exceptionally well and mistakes are legitimately repaired. The application works as indicated by the imperatives given in the individual framework. Promote upgrades and improvements can be made to the application so that the application capacities seem alluring and in a more helpful way than the present one.

Each application has its own faults. The venture has secured every one of the prerequisites. Advance prerequisites and upgrades should effortlessly be possible since the coding is for the most part organized and particular in nature. Changing the current modules or including new modules can affix enhancements. Promote improvements that can be made to the application are –

1) Better UI (User Interface) can be given to the program.
2) A better architecture can be chosen so as to support the functioning of large number of nodes.
3) A better way of loading the sound files can be administered so as to increase the performance.
4) The motion of the nodes can be mapped according to the perlin nose in 2D so as to provide a more fluid motion of nodes over the space.
5) The code can be provided with more modularity which will include well-defined functions with a better design pattern.
6) The nodes in the system can be provided with a better battery draining algorithm while being attacked.
7) The application can be provided a mechanism for evenly distributing the nodes and avoid clustering of nodes in the network using a probability based approach.

References

[1] Samira Akhbarifar1, A. M. Rahmani2, “A Survey on key pre-distribution Schemes for security in Wireless Sensor Networks”, Research and Science University, Tehran, Iran (2014).

[2] Rohith Singi Reddy, Key Management in Wireless Sensor Networks Using a Modified Blom Scheme”, (2011).

[3] Jing Deng & Pramod K. Varshney, “A Pairwise Key Predistribution Scheme for Wireless Sensor Networks”, (2005).

[4] Abdalkahik W. Hussain, Mahmood K. Ibrahem, ” , An Efficient Pairwise and Group Key Management Scheme For Wireless Sensor Network”, (2015), pp: (25-31).

[5] Prof. N. D. Kale & Aher Nisha N., “A Survey on key Generation and Pre-distribution Technique in wireless Sensor Network”, (2014).

Appendices

Code

```
var broadcast="Network Secure";
var flag=false,mflag=false,cflag=false;
var r1=[],r2=[];
var range=200;
var nodeid=0,did=0;
var time=0.01;
var balls=[],malicious=[],maliciousneighbours=[];
var total=5;
var count=0;
var s1,s2;
var selected=0;
var lamda,s,G,D,m,t,f,bg,alertsound,normalsound;

function preload(){
  alertsound=loadSound('alertsiren.mp3');
  //normalsound=loadSound('normal.mp3');
}

function setup() {
//normalsound.loop();
//lamda=computeLamda(total);
  //s=2;
  //G=computePublicMatrix();
  //printPublicMatrix(G);
  //D=computePrivateMatrix();
  //printPrivateMatrix(D);
  //m=multiply(D,G);
  //t=transposeMatrix(m);
  //f=multiply(t,G);
  //display(f);
  createCanvas(600,600);
  background(0);
  insertNodes();
}

function draw() {

  if(mflag){
    //normalsound.stop();
    alertsound.play();
  }
  else
    alertsound.stop();
  //normalsound.play();

  background(0);

  if(malicious.length==0)
    mflag=false;

  if(mflag==true)
    background(200,0,0,100);
  else
    background(0,0,200,0);

  for(var i=0;i<balls.length;i++){
    balls[i].move();
    balls[i].display();
  }
```

```
if(flag===true){
  for(var i=0;i<malicious.length;i++){
    malicious[i].move();
  malicious[i].display();
}
}

if(selected==1){
  computeInRange(s1);
}
if(mflag==true)
  background(200,0,0,100);

if(selected==2){
  if(dist(s1.x,s1.y,s2.x,s2.y)>=range){
      stroke(255,0,0);
      line(s1.x,s1.y,s2.x,s2.y);
  }
  else{
      stroke(100,205,0);
      line(s1.x,s1.y,s2.x,s2.y);
  }
  computeInRange(s1);
  computeInRange(s2);

  if(mflag==true)
  background(200,0,0,100);
  else
  background(0,0,200,0);
  }
```

```
    if(malicious.length>0 && mflag==true){
      //var abc=[];
      for(var i=0;i<malicious.length;i++){
      computeInRange(malicious[i]);
        }
        if(mflag==true)
    background(200,0,0,100);
         else
    background(0,0,200,0);
    stroke(0);
  }

}

function keyTyped() {

  if (key === 's') {
    spawnMaliciousNodes(3);
    flag=true;
  }

  if (key === 'l') {
    mflag=true;
  }

if (key === 'b') {
      console.log(broadcast);
  }
```

```
if (key === 'm') {
    for (var i = 0; i <malicious.length; i++) {
      console.log("Malicious id:"+malicious[i].id);
    }
  }

  if (key === 'd') {
   if(bflag==true){
      for (var i = 0; i <malicious.length; i++) {
      if(malicious[i].id==did){
        malicious.splice(malicious.indexOf(malicious[i]), 1);
      }
    }
   }
  }

}

function mousePressed(){

  for(var i=0;i<balls.length;i++){
    var d=dist(balls[i].x,balls[i].y,mouseX,mouseY);
    if(d<=balls[i].r*2){
      if(selected==0){
        s1=balls[i];
        balls[i].a=255;
        balls[i].b=0;
        r1=computeInRange(s1);
        ellipse(balls[i].x,balls[i].y,balls[i].r*2,balls[i].r*2);
        selected++;
        break;
      }
      else if(selected==1){
        s2=balls[i];
        balls[i].a=255;
        balls[i].b=0;
        r2=computeInRange(s2);
        ellipse(balls[i].x,balls[i].y,balls[i].r*2,balls[i].r*2);
        selected++;
        break;
      }
    }//end if

  }//end for
}

function Ball(x,y,type){
  this.id=++nodeid;
  this.x=x;
  this.type=type;
  this.y=y;
  this.xoff=random(50),this.yoff=random(50,100);
  this.inbox=[];
  this.outbox=[];
  this.w=255;
  this.r=16;
  this.x=constrain(x,this.r,width-this.r);
  this.y=constrain(y,this.r,height-this.r);
  this.battery=this.r*2;
  this.a=100;
  this.b=255;
  this.send=function(txt,receiverid){
  this.outbox.push([receiverid,txt]);
  }

  this.receive=function(txt,senderid){
  this.inbox.push([senderid,txt]);
  this.inbox.splice(100,this.inbox.length-1);
  this.checkAuthenticity(senderid);
  }

  this.checkAuthenticity=function(senderid){
    //console.log(inbox[0].senderid);
  if(this.inbox.length>0){
    console.log(senderid);
    if(senderid>total){
      broadcast="Mallicious Node detected with senderid :"+senderid;
      bflag=true;
      did=senderid;
    }
    else
    {
      //communicate
    }
//console.log(this.broadcast);
  }
  }

  this.display=function(){
    smooth();
    fill(this.a,0,this.b);
    noStroke();
    ellipse(this.x,this.y,this.r*2,this.r*2);
    fill(this.w);

    if(this.type==0)
    this.battery-=time*0.01;
    else
        this.battery-=time;



    if(this.battery<0)
      this.battery=0;

    ellipse(this.x,this.y,this.battery,this.battery);

    fill(200, 2);
  rect(0, 0, width, height);
  // write the text in black and get its bounding box
  fill(100);
  var
t="id:"+this.id+"\nx:"+Math.floor(this.x)+"\n"+"y:"+Math.floor(this.y)+"\n"+"battery:"+(
Math.round(this.battery * 100) / 100);
        text(t,this.x-10,this.y+30);
          stroke(255);
    fill(120,-12);
    };

  this.move=function(){
    this.x=map(noise(this.xoff+=0.001),0,1,0,width);
    this.y=map(noise(this.yoff+=0.001),0,1,0,height);
    this.x=constrain(this.x,this.r,width-this.r);
  this.y=constrain(this.y,this.r,height-this.r);
  };

}

function establishLink(b1,b2){
  var txt = "crypto";
  //console.log(b1.id);
  //var txt = crypto.randomBytes(20).toString('hex');
  b1.send(txt,b2.id);
  b2.receive(txt,b1.id);
};

function computeInRange(s){

  fill(255,80);
  ellipse(s.x,s.y,range*2,range*2);
    var r=[];
  for(var i=0;i<balls.length;i++){
      if(s==balls[i])
        continue;
      else{
          if(dist(balls[i].x,balls[i].y,s.x,s.y)<=range){
          r.push(balls[i]);
          //stroke(20,0,200 );
          if(mflag==true){
            //console.log(s);
            establishLink(s,balls[i]);
          }
          line(balls[i].x,balls[i].y,s.x,s.y);
      }
      }
  }
  fill(255,80);
  return r;
}

function insertNodes(){
    var xval=random(width);
    var yval=random(height);
  //  var b=new Ball(xval,yval);
  while(count<total){
    var overlapping=false;
    xval=random(width);
    yval=random(height);

      for(var j=0;j<balls.length;j++){
        {
            if((dist(xval,yval,balls[j].x,balls[j].y)<=balls[j].r+balls[j].r) || (xval<balls[j].r ||
yval<balls[j].r || xval>width-balls[j].r|| yval>height-balls[j].r)){
                overlapping=true;
              break;
            }
        }
      }

    if(!overlapping){
      b=new Ball(xval,yval,0);
      balls.push(b);
      count++;
    }
```

```
}
}



function spawnMaliciousNodes(n){
  var xval=random(width);
    var yval=random(height);
  //  var b=new Ball(xval,yval);
  var mcount=0;
  while(mcount<n){
    var overlapping=false;
    xval=random(width);
    yval=random(height);

      for(var j=0;j<malicious.length;j++){
        {
            if((dist(xval,yval,malicious[j].x,malicious[j].y)<=malicious[j].r+malicious[j].r) ||
(xval<malicious[j].r || yval<malicious[j].r || xval>width-malicious[j].r|| yval>height-
malicious[j].r)){
                overlapping=true;
              break;
            }
        }
      }

    if(!overlapping){
      b=new Ball(xval,yval,1);
      b.a=100;
      b.b=100;
      b.w=0;
      malicious.push(b);
      mcount++;
    }
}
}



function computeLamda(total){
    var x=total;
    while(!isPrime(x))
      x--;
  return x;
  }

function isPrime(value) {
    for(var i = 2; i < value; i++) {
        if(value % i === 0) {
            return false;
        }
    }
    return value > 1;
}

function computePublicMatrix(){
var G= new Array(lamda);
for (var i = 0; i <= lamda; i++) {
  G[i] = new Array(total);
    for(var j=1;j<=total;j++){
        G[i][j-1]=Math.pow(Math.pow(s,j),i);
    }
}
```

```
  return G;
}

function printPublicMatrix(G){
  console.log("2D public:\n");
for (var i = 0; i <= lamda; i++) {
  var str="";
    for(var j=0;j<total;j++){
      str+=(G[i][j]+" ");
    }
  console.log(str);
}
}

function computePrivateMatrix(){
  var D= new Array(lamda);
for (var i = 0; i <= lamda; i++) {
  D[i] = new Array(lamda);
    for(var j=0;j<=lamda;j++){
        if(i==j)
          D[i][j]=Math.floor(random(500));
        else if(i<j){
          D[i][j]=Math.floor(random(1000));
        }
      else if(i>j)
          D[i][j]=D[j][i];
    }
}
  return D;
}
```

```
function printPrivateMatrix(D){
  console.log("2D private:\n");
for (var i = 0; i <= lamda; i++) {
  var str="";
    for(var j=0;j<=lamda;j++){
      str+=(D[i][j]+"  ");
    }
  console.log(str+"\n");
}
}

function display(m) {
  for (var r = 0; r < m.length; r++) {
    console.log(" "+m[r].join(' ')+" \n");
  }
}
```