An Efficient Novel Key management scheme using NchooseK algorithm for Wireless Sensor Networks

Harjot Bawa[1], Parminder Singh[2] and Rakesh Kumar[3]

[1] Research Scholar, Department of Information Technology, CEC Landran
harjotbawa@ymail.com
[2] Assistant Professor, Department of Information Technology, CEC Landran
singh.parminder06@gmail.com
[3] Associate Professor, Department of Computer Science, Sachdeva Engg. college for Girls, kharar
rakesh77kumar@yahoo.com

Abstract

In Wireless Sensor Networks Key Management is a very challenging phenomenon. This paper gives an illustration and demonstration of mathematical model of new key management scheme which overcomes the limitation of Pre-Shared key scheme, which is extensively used in wireless sensor networks by most of the vendors. The environment of WSN is challenged by many limitations due to which there is an urgent need to manage memory which is consumed while provisioning of the key in the wireless sensor network by using N-chooseK algorithm. We were successfully able to build a more reliable network in terms of network connectivity .Secondly it offers a scalable solution as there is no need to keep key material of all the network in each sensor node.

Keywords

Wireless Sensor Network(WSN), Pre-Key Distribution Scheme(PSK), NchooseK Algorithm

1. INTRODUCTION

1.1 Key Management in WSN

Key is the most important component for most of the Cryptographic algorithms. Keys are generally numbers randomly selected from a large set of numbers. Management of these keys are very important in cryptography. Management of keys include the following:

1. Key Generation : It is the process in which a pool of keys are generated. It can be done in offline or online mode by a trusted authority or automated algorithm.

2. Key Establishment : It is the most important phase of key management process. Key establishment is the process by which right keys for right users (sensors) can be determined and key rings for each users are sent to them accordingly.[10]

Key establishment can be done in many ways. Trusted Authority can help in sending the keys to each user through a secure channel. But this mechanism is a costly one and does not suit for sensor networks. So, in sensor networks Key Pre-distribution is used in which key rings are installed in the nodes before deployment of network in offline mode.[1]

Key establishment process in Wireless sensor networks mainly consists of three phases.

1. Key pre-distribution : Pre-loading keys in sensor nodes prior to deployment. The keys present in a sensor node constitute the key ring of the sensor.

DOI : 10.5121/ijcnc.2012.4610

2. Shared key discovery : To find a common shared key between two communicating nodes.

3. Path key establishment : If a common key does not exists, then a path has to be found between the communicating nodes. A path key is then established between the communicating nodes.

In Key Pre-Distribution scheme, secret keys are placed in sensor nodes before deployment. When the nodes are deployed over the target area, the secret keys are used to create the network.

1.1.1 Key pre-distribution

The important key pre-distribution schemes which are highly used can be classified as follows:
• Probabilistic key pre-distribution scheme.
• Polynomial-based key pre-distribution schemes [5].
• Blom's matrix-based key pre- distribution schemes.
• Deterministic key pre-distribution schemes.
During the phases of Key Pre-distribution , secret keys are generated, placed in sensor nodes, and each sensor node searches the area in its communication range to find another node to communicate.[5]

2.KEY MANAGEMENT SCHEMES SUMMARY

2.1 Eschenauer and Gligor's method [1]

Eschenauer and Gligor's method is the first key distribution scheme especially designed for sensor networks. It also constitutes the foundation of the subsequent key distribution schemes in sensor networks

Before sensor deployment, a key pool P of p distinct keys with key identifier is randomly generated. For each sensor node si, a subset Ri of r keys with their key identifiers is randomly chosen from P. This subset Ri is also called the key ring of sensor node si. After sensor deployment, two sensor nodes with (at least) one common key in their key rings can use this common key as the shared key. In the literature, this procedure of discovering the common key in two key rings is often called shared key discovery.

After the shared key discovery, if two sensor nodes do not have the common key in their respective key rings, they resort to a procedure called path key establishment. The goal of path key establishment is to find a sequence of secure links, which is defined as the communication links whose two ends have found their shared key in shared key discovery. Once path key establishment is successfully finished, that is, a sequence of secure links has been achieved between two sensor nodes that cannot find their shared key in shared key discovery, these two sensor nodes can establish their shared key by, for example, sending the shared key from one end to the other end. During the transmission, as each link is the secure link, the confidentiality of shared key between two ends can be guaranteed.[6]

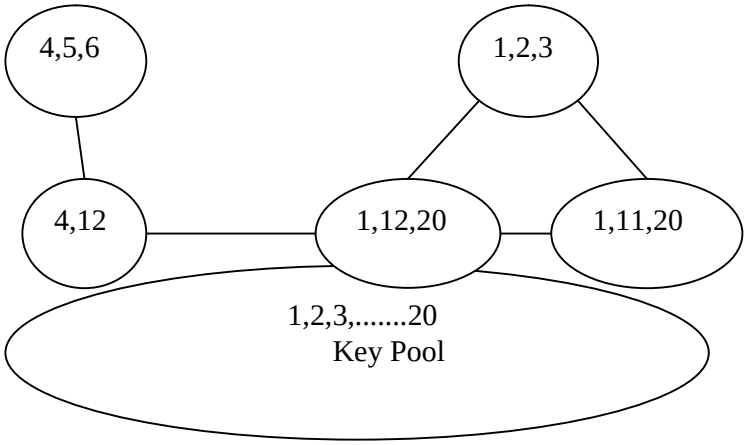


Figure.1 Example of Eschenauer and Gligor's method [6]

2.2 q-Composite key pre-distribution scheme [3]

q-Composite scheme can be thought of as a natural extension of Eschenauer and Gligor's method. Its security enhancement is mainly due to the use of multiple keys, instead of single key in Eschenauer and Gligor's method. After sensor deployment, the shared key discovery and the path key establishment are also the same as those in Eschenauer and Gligor's method. The only difference is that, in q-composite scheme, q common keys in the key rings, instead of a single common key, should be found to construct the shared key.

For both the EG and the q-composite schemes, if a small number of sensors are compromised, they may reveal to a large fraction of pair wise keys shared between non-compromised sensors.[2]

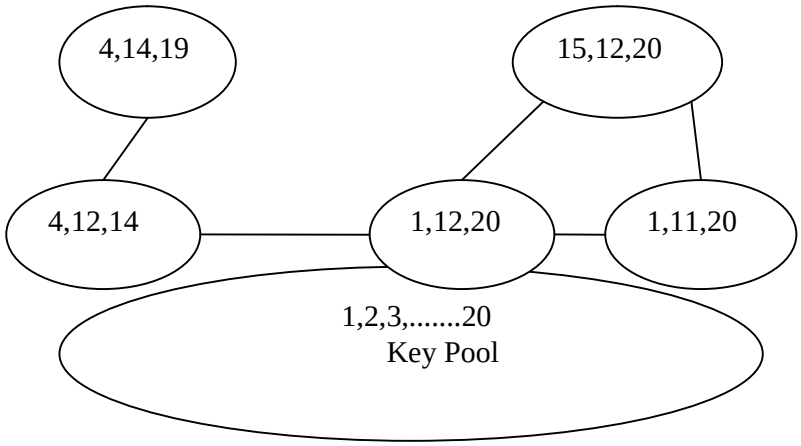


Figure.2 An example of q-composite key pre-distribution scheme

2.3 Method by Blundo et al. [9]

Assume that the authority randomly selects a bivariate t-degree symmetric polynomial. The symmetric polynomial possesses a property of f(x, y) = f(y, x). For sensor node si, a polynomial share f (i, y), which is a univariate t-degree polynomial, is calculated. Then, f(i, y) is stored in sensor node si. After sensor deployment, the key can be obtained by sensor node si by calculating f (i, j) as long as sensor node si would like to have a shared key with sensor node sj.

For sensor node sj, similar procedures can be conducted by sensor node sj; that is, f (j, i) is computed. Because of the property of f(x, y) = f(y, x) in the underlying symmetric polynomial, their calculated keys should be the same and can be the shared key.

2.4 Random perturbation‐based key establishment scheme [7]

The method by Blundo et al. can guarantee perfect connectivity. However, its resilience against sensor compromises is not considered to be acceptable because once a fixed number of sensor nodes have been compromised, the security of the entire sensor network will suddenly crash. To address this issue while preserving perfect connectivity, a random perturbation‐based key distribution scheme is proposed. In essence, certain random perturbations are introduced into the method by Blundo et al. Because of the added random perturbation, the original shared key is gone. However, some portions of destroyed keys can be extracted by the sensor nodes and can be used as the shared key. Before sensor deployment, a bivariate t‐degree symmetric polynomial is randomly generated as in the method by Blundo et al. Unlike the method by Blundo et al., the perturbation polynomials for each sensor node are further generated. The generation of perturbation polynomial is not totally random, and has to follow the rule that adding the perturbation into the univariate polynomial generated from the bivariate t‐degree symmetric polynomial will not lead to the fluctuation of coefficients. In particular, assume that a bivariate t‐degree symmetric polynomial f(x, y) is chosen, and the perturbation polynomial φi(y) is chosen for sensor node si. Then, instead of f(i, y), f(i, y) +φ(y) is stored in sensor node si. However, because of the addition of φ(y).

3. MOTIVATION

In a Wireless Sensor Network (WSN for short), individual sensor nodes, or sensors, are constrained in energy, computing, and communication capabilities. Typically, sensors are mass-produced anonymous commodity devices that are initially unaware of their location. Once deployed, sensors should self-organize into a network that works unattended .Due to the fact that individual sensor nodes are anonymous and that communication among sensors is via wireless links, sensor networks are highly vulnerable to security attacks that leads to usage of more memory and overhead in the gateways and nodes and basic mechanism used to secure them is by using some Key management scheme.

Typically in Pre-Shared Key type of schemes ,there is a major drawback related to its distribution property. As all the secret information must be preloaded for any further exchange of information which would ultimately lead to secure client-server (sensor) connection in which keys are already known. Another drawback is the overall resilience as one element will store all the Pre-Shared keys, such element is the weakest link of the security chain, and any adversary that gains control of that element will take control of the whole network.

In order to overcome above stated limitations we are using a proposed scheme which uses the Nchoosek algorithm.
The Pre Shared Key scheme does not offer any option to really upgrade ,reset, or scale up keys in case of any adversary .Moreover ,the network may suffer in connectivity due to large randomly generated keys reading large pool of disjoint sets of keys.

3.1 Objectives of Research

1. Develop a simulated environment of Wireless Sensor Network based on IEEE 802.15.4 protocol stack.

2. Develop a key management protocol based on Random Key Polynomial distribution scheme.
3.Evaluate performance in terms of Key connectivity and delay.
4.Based on the evaluation develop a new Key Management protocol to overcome limitations.
5.Draw comparative evaluation of both Key Management schemes.

3.2 Methodology

It deals with the simulation study of the routing protocol in Wireless Sensor Networks. The procedure used for simulations is explained and the scenarios for which simulations are carried out are described. The observations of the simulation study are plotted as graphs and conclusions are carried out on the basis of these graphs.

Simulation is the research tool of choice for majority of researchers. The deployment of wireless applications or protocols in the context of Wireless Sensor Network also requires stepping through simulation phase.
There are more than twenty simulators available in which Ad-hoc network simulations. Various simulators are Ns-2, GlomoSim, OPNET, MatLab, Qualnet etc. To use any simulator following points were kept in consideration.

- Identify the appropriate simulator supporting the proposed research work and current version of simulator.
- Identify the compatibility of particular version of simulator with operating system available.
- Identify various settings of variables to use a simulator.

3.3 Proposed Work

The flow chart in the below Figure depicts the various steps which are carried out during the Pre-key distribution process. The simulation is carried out using the Network simulator (version 2.35), which simulates the events such as sending, receiving, dropping, forwarding, etc. The wireless channel is used as the sensor nodes deployed communicate wirelessly with each other. The propagation models are used to compute the received power. When a packet is received, the propagation model determines the attenuation between transmitter and receiver and computes the received signal strength. The two-Ray ground Radio propagation model is used. An omni-directional antenna is employed for carrying out the transmissions which can transmit signal over a 360 degree angle. Omni-directional wireless sensor networks are modelled such that a bidirectional link is established between neighbouring sensor nodes if they are within communication radius [4].

The simulation is carried out using the Network simulator(version 2.35),which simulates the events such as sending, receiving, dropping, forwarding, etc. The wireless channel is used as the sensor nodes deployed communicate wirelessly with each other. The propagation models are used to compute the received power. When a packet is received, the propagation model determines the attenuation between transmitter and receiver and computes the received signal strength. The two-Ray ground Radio propagation model is used. An omni-directional antenna is employed for carrying out the transmissions which can transmit signal over a 360 degree angle. Omni-directional wireless sensor networks are modelled such that a bidirectional link is established between neighbouring sensor nodes if they are within communication radius. [4] The scenario is simulated for 150 seconds. The participating nodes are mobile. The routing protocol which monitors and carries out the transmission is Ad-hoc On Demand Distance Vector routing Protocol(AODV).

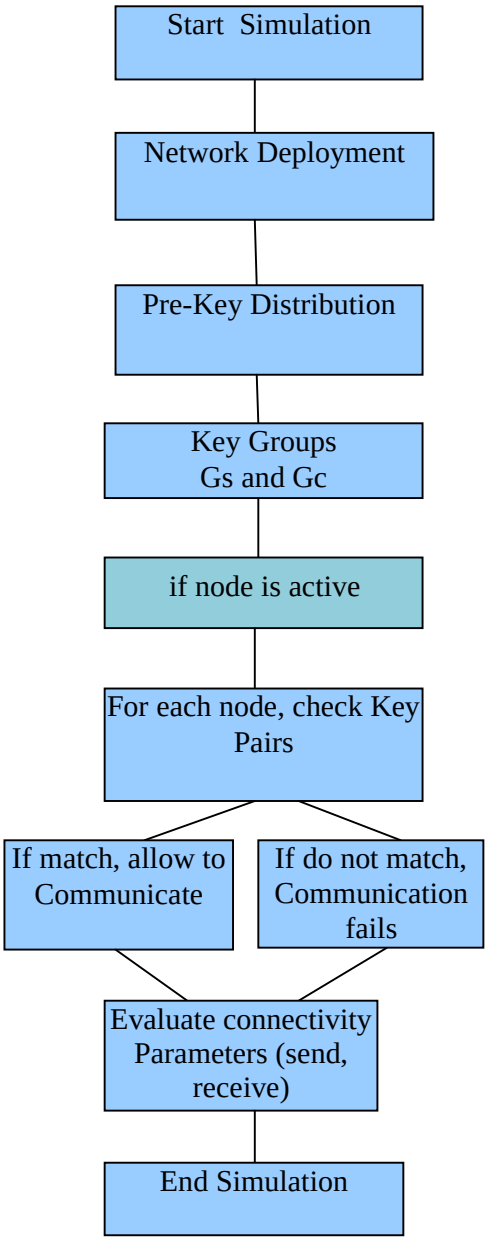


Figure.3 Flow Diagram

The following table gives an overview of all the simulation parameters used.

Table 1.Simulation Parameters

Parameter	Value
Simulator	NS-2.35
Channel Type	Wireless Channel
Mobility Model	Two-Ray ground Radio Propagation Model
Network Interface Type	Wireless Phy/IEEE 802.15.4
Antenna Model	Omni-directional
Number of mobile-nodes	50
Routing Protocol	AODV
Simulation Time	150 sec
Simulation area (m*m)	1000 *1000
Packet Size	1024 bits

(I) In pre-shared key scheme, the manufacturer of the sensors normally gives predefined keys to each sensor. This group of keys is denoted by G. This group has a concept of having keys for client and server. So therefore, for a successful communication among these, there has to be a pre-shared key between client and server. In this each client must also store the key from the server, so in case of scalability we need to be very careful to buy only those sensors which have pre-configured keys and which have a common key with the server and belongs to a particular G-group.
Typically when these pre-configured keys are distributed among sensors, they are created on the basis of pseudo random algorithm in which no case is taken whether these keys will finally have a certain level of connectivity in a sense that they will be helpful in securing a network but at the same time having large number of key sets which are an intersection of the key group pairs. So therefore it reaches a point sometimes that inspite of the fact these sensors are in a comfort zone to communicate with each other.

(II) Key groups- Mathematical model of Pre-shared key
1. let S be the number of sensors to be deployed in the network.
2. let S_{sn} be the number of servers which are to be deployed as sensors.
3. let S_{Cn} be the number of sensors which are to be deployed as clients in a network.
4. Let G be the group of keys allotted to a network.
5. Let the group be bifurcated into G_c and G_s representing keys that belong to server nodes and client nodes.
6. Since the number of servers will be less and number of clients will be more, the groups need to be allotted accordingly.

- Disjoint sets

Two sets are said to be disjoint if both the sets have nothing common among them.

let there be two sets of keys Gs and Gc.
The set of Gs and Gc is said to be disjoint , if the keys in these two sets do not match at all.

If , $Gs= \{PSK_{sk1}, PSK_{sk2},.....PSK_{skn}\}$

$Gc=\{PSK_{ck1},PSK_{ck2},.....PSK_{skn}\}$

then , Probability $(Gs \eta Gc) = \varphi$ or

Probability <0.2

The intersection of these two key sets is a null value or it's a very small value. Thus in case of disjoint sets the connectivity represented by the number of packets received at the other end is less.

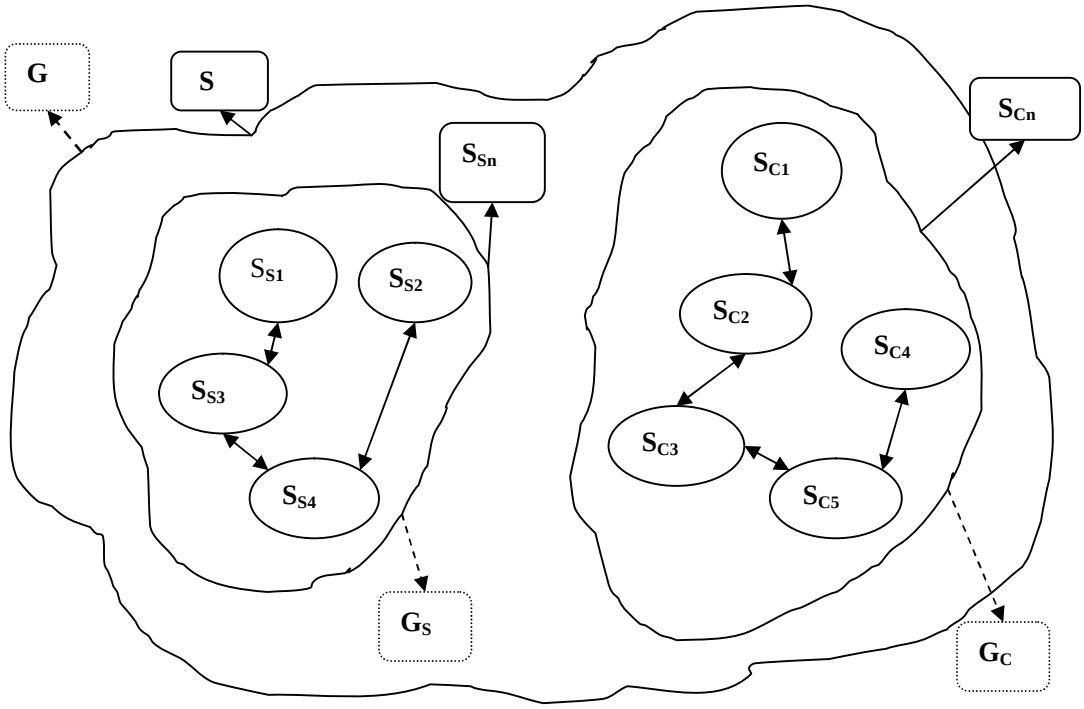


Figure.4 Wireless Sensor Network

In a wireless sensor network, there is a set of client sensors and server sensors. The client and server sensors are organized into clusters as shown in the figure 3.5 above.
These server sensors and client sensors are communicating with each other on the basis of key match between them. If the key match occurs between the server and the client sensor, then the communication takes place otherwise there's an authentication failure.
In a Pre-Shared scheme, as there are sets of server and client sensors. Each server sensor stores the key information of each client sensor and each client sensor stores the information of the each server sensor. Each sensor has maintained a key pool with it. When a node wants to communicate with another node, it checks for the availability of the keys in its key pool. If there's a match of keys, the communication occurs otherwise it fails.

If there exist a network with 1000 server sensors and 1000 client sensors, then the communication overhead of each node would increase to a greater extent. As each client sensor needs to store the information of the rest 999 server sensors in the network and each server sensor needs to store the information of the remaining 999 client sensors. Also the Wireless

Sensor Network is an ad-hoc network, which further leads to an increase in the overhead of the individual nodes in the network and thus reduces the performance of the network.
As the sensors are memory constrained devices which means that they would become dead at regular intervals of time which would lead to an authentication failure. Under this scheme only 10% of the communication takes place and rest are the failures.
In order to overcome this problem and reduce the memory overhead of the sensors, we are making use of computational and combinational algorithms. These lead to an increase in the performance of the network as the sensors need not to store the keys which ensure less memory overhead and reduced authentication failures which occur due to key mismatch.

Once the simulation starts, the sensors are deployed in the network. The network is then bifurcated into server sensors and the client sensors. As the sensors do not have the pre-configured keys, the NchooseK is making combinations of keys and distributing them to the server sensors and the client sensors.

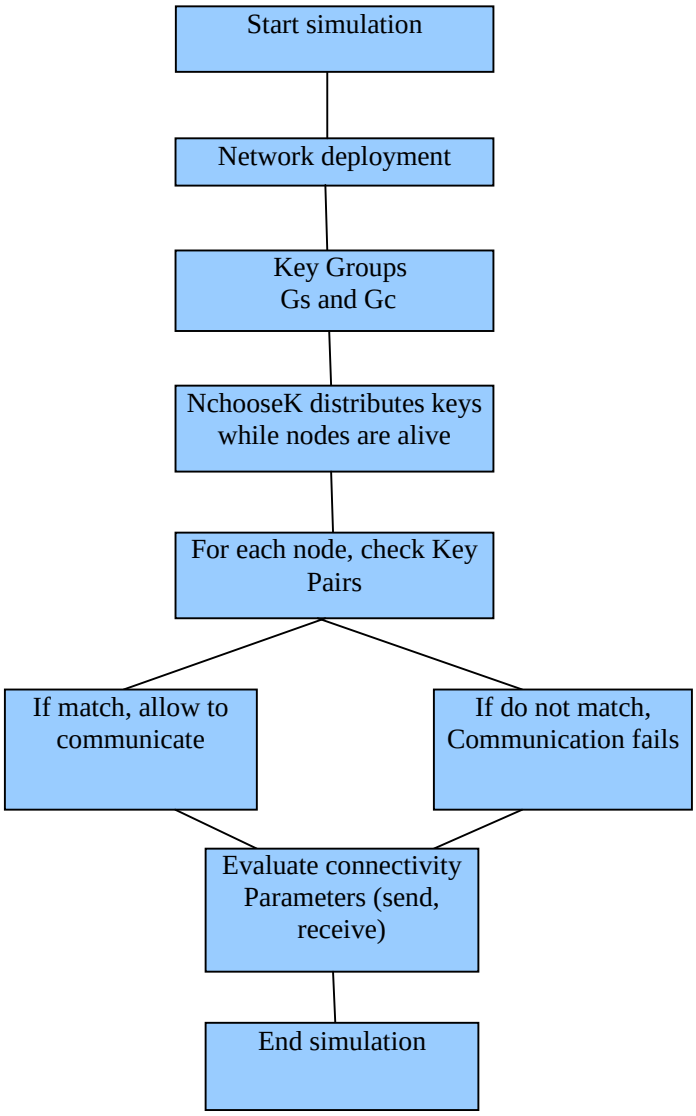


Figure.4 Flow Diagram

In a wireless sensor network, in order to ensure a secured peer to peer communication, a session key establishment is necessary. The protocol either uses some trusted authority to distribute the keys or pre-distribute the communication keys among the sensor nodes.
The key pre-distribution is difficult to achieve as there are thousands of nodes deployed in the wireless sensor network and the session keys need to be stored in every node. These sensor nodes are constrained in memory and they do not have enough space to store the keys.
So here in our work, the Adaptive Random Pre-distribution scheme is used. This scheme is divided into two parts namely the Key Pool and the selection algorithm. The key pool is used to store the randomly generated keys and the selection algorithm is used to select the keys from the key pool. The selection algorithm which selects the keys and enables the communication between the keys is the NchooseK algorithm. The nodes that have a common key are securely able to communicate with each other with the help of shared key.

3.4.1 Explanation of NchooseK algorithm

N-choose-k is defined to be the number of ways one can select k keys from a pool of n keys.[8]

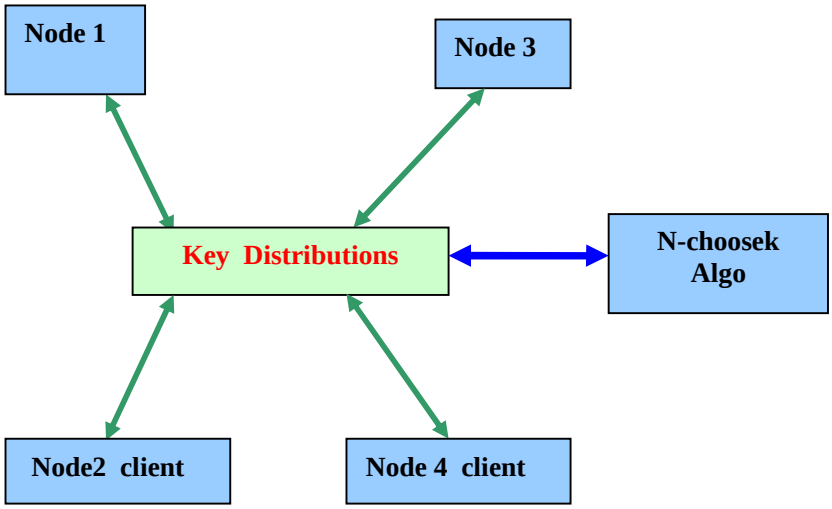
$$\text{n choose k} = \frac{n!}{(n-k)!\,k!}$$

r = rand(n) returns an n-by-n matrix containing pseudo-random values drawn. The sequence of numbers produced by rand is determined by the internal settings of the uniform random number generator

C = nchoosek(n,k) where n and k are nonnegative integers, returns $n!/((n–k)!\ k!)$. This is the number of combinations of n things taken k at a time.

C = nchoosek(v,k), where v is a row vector of length n, creates a matrix whose rows consist of all possible combinations of the n elements of v taken k at a time. Matrix C contains $n!/((n–k)!\ k!)$ rows and k columns.

3.4.2 Key Selection and algorithm



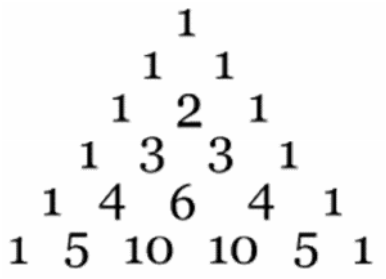


Table.2 Server Key Pool

Keys Nodes	1st	2nd	3rd	4th	5 th	6 th	7th	8 th
N1	1	1	2	2	6	6	0	0
N2	1	1	3	3	10	10	0	0
N3	1	1	4	4	0	0	0	0
N4	1	1	5	5	0	0	0	0

Table.3 Client Key Pool

KEYS Nodes	1st	2nd	3rd	4th	5 th	6 th	7th	8 th
N1	1	0	0	0	0	0	0	0
N2	0	1	0	0	0	0	0	0
N3	0	0	0	0	0	1	0	0
N4	0	0	0	1	0	0	0	0

Example of Nchoosek keys

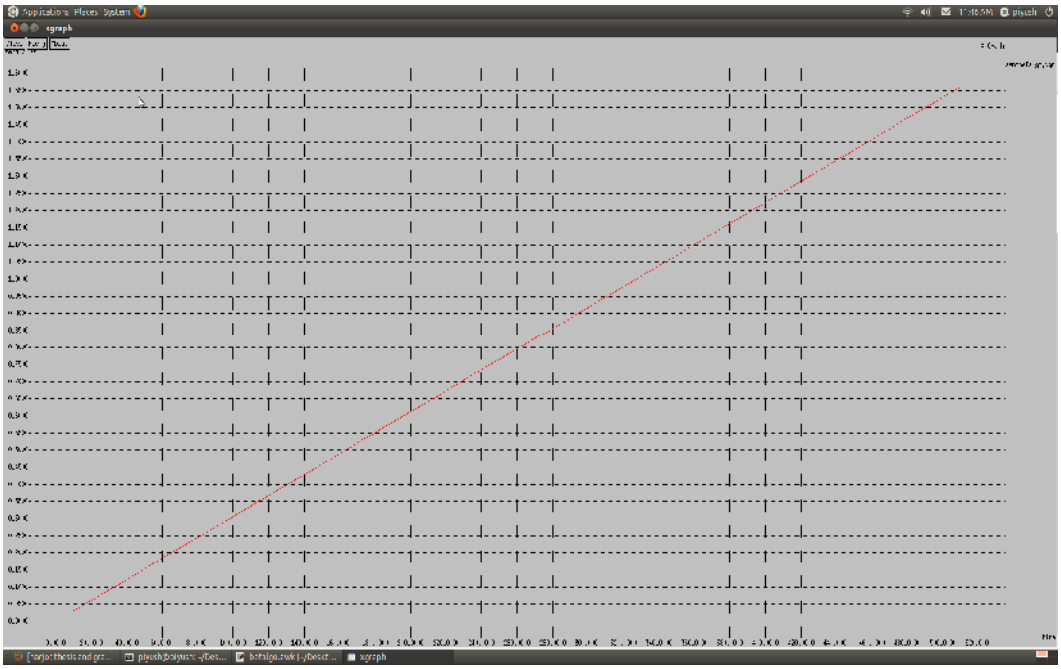
2	**4**	**6**	**8**	**10**	**12**	**14**	**16**
2	**4**	**6**	**8**	**10**	**12**	**14**	**18**
2	**4**	**6**	**8**	**10**	**12**	**14**	**20**
2	**4**	**6**	**8**	**10**	**12**	**16**	**18**
2	**4**	**6**	**8**	**10**	**12**	**16**	**20**
2	**4**	**6**	**8**	**10**	**12**	**18**	**20**
2	**4**	**6**	**8**	**10**	**14**	**16**	**18**
2	**4**	**6**	**8**	**10**	**14**	**16**	**20**
2	**4**	**6**	**8**	**10**	**14**	**18**	**20**
2	**4**	**6**	**8**	**10**	**16**	**18**	**20**
2	**4**	**6**	**8**	**12**	**14**	**16**	**18**

4 .SIMULATION RESULTS

4.1 The following graphs were obtained before the implementation of NChooseK algorithm.

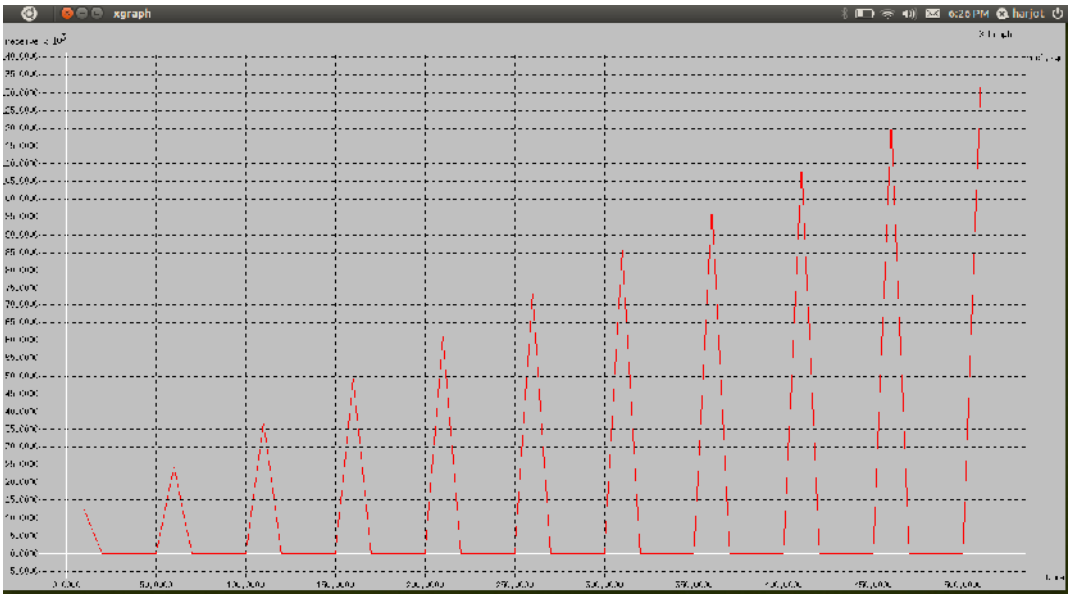
4.1.1 Throughput of sending packets

The first graph shows that there is a steady increase of request sent to the other nodes starting from default source to the destination. It can be seen from the graph that with the passage of time, the rate of increase is exponential in nature .Since more and more nodes start communicating and sending request to each other .It's at this stage of sending HELLO message, which confirms that request has been sent to the nodes and they want to communicate with each others.



4.1.2 Throughput of receiving packets

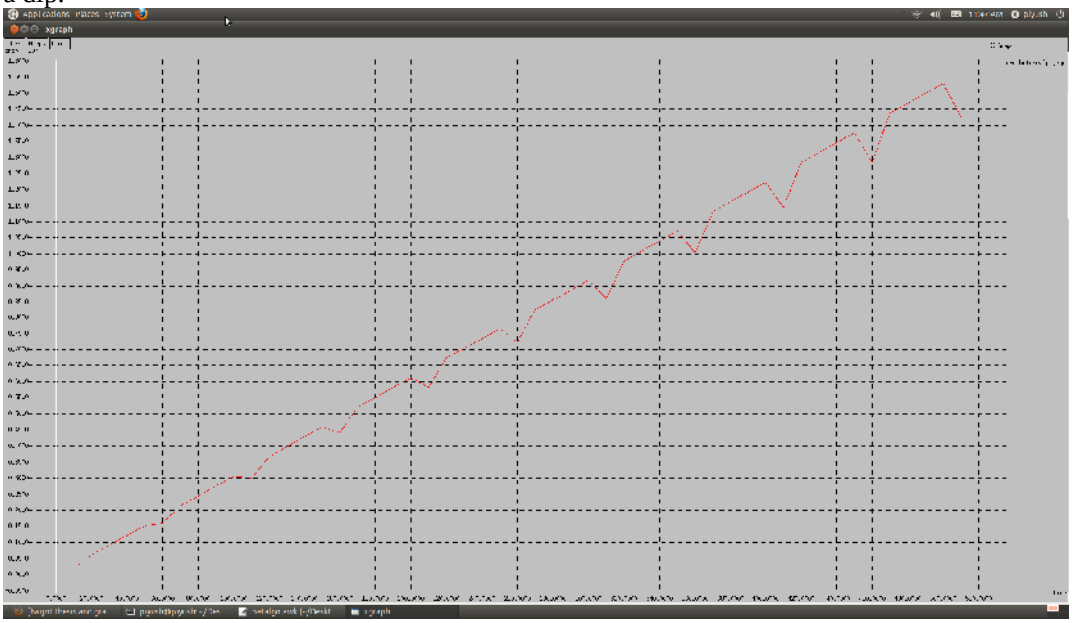
It's apparent from the graph that as the packets are sent ,they are received but there is a sudden drop as the simulation proceeds. This means that at this stage the Key Pre-Distribution scheme works. But due to some resilience, there is a sudden drop of packets in the network. In other there is an implementation of disjoint sets. This is attributed to the fact that inspite of nodes being in the same range and having minimum cost of route as well as have established message routing protocol cycle complete, it suddenly shows drop in receiving data .As the key pairs seem to be not matching with each other and therefore they do not receive data and there's a dip. Although with the passage of time there's always an increase in peaks and valleys. In the graph the packets received are shown by the graphs and the packets dropped are shown by the valleys. The graph illustrates that it has high computational overhead which is not too much suitable for WSN due to effect of resilience and require more memory for storage of keys as well as exchange of keys for authentication.



4.1.3 Drop

The drop analysis depends on the send and receive graphs. Mathematically, drop is the difference between packets which are sent and received during a time span. As seen from the graph that with the passage of time if the packets are being received normally then there's a steady increase in the straight line but the packets are dropped, it shows a dip in the graph.

The drop here (PSK) is due to large number of pre-configured keys which are random in nature and may not lead to a common set of key pairs which would have ensured the connectivity. It is apparent from the graph when the communication starts, as the keys are exchanged when two nodes as per the protocol can communicate are unable to communicate as per their keys are disjoint in nature. So, more and more communication occurs ,after some steady increase there is a dip.

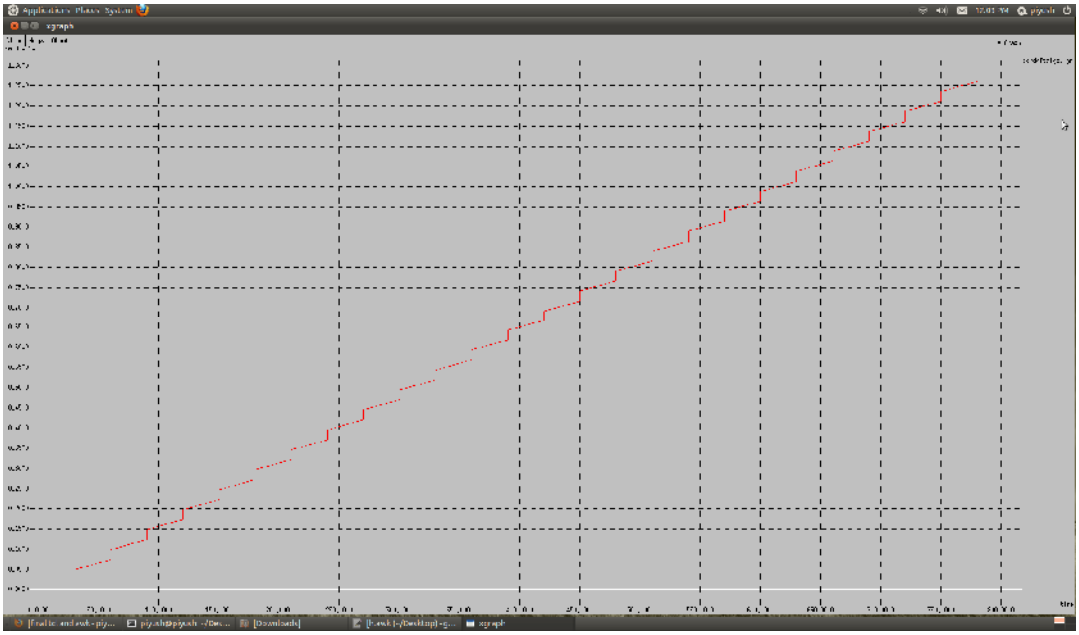


The graph also explains that the packet delivery ratio has not proper connectivity with respect to scheme known as PSK. The probability against number of keys required is not proper which reduces the probability of connectivity.

4.2 Following are the graphs obtained after the implementation of the algorithm .As it can be seen from the graphs that there is a sufficient reduction in the drop as compared to the graphs which were obtained before the implementation of the NChooseK algorithm.

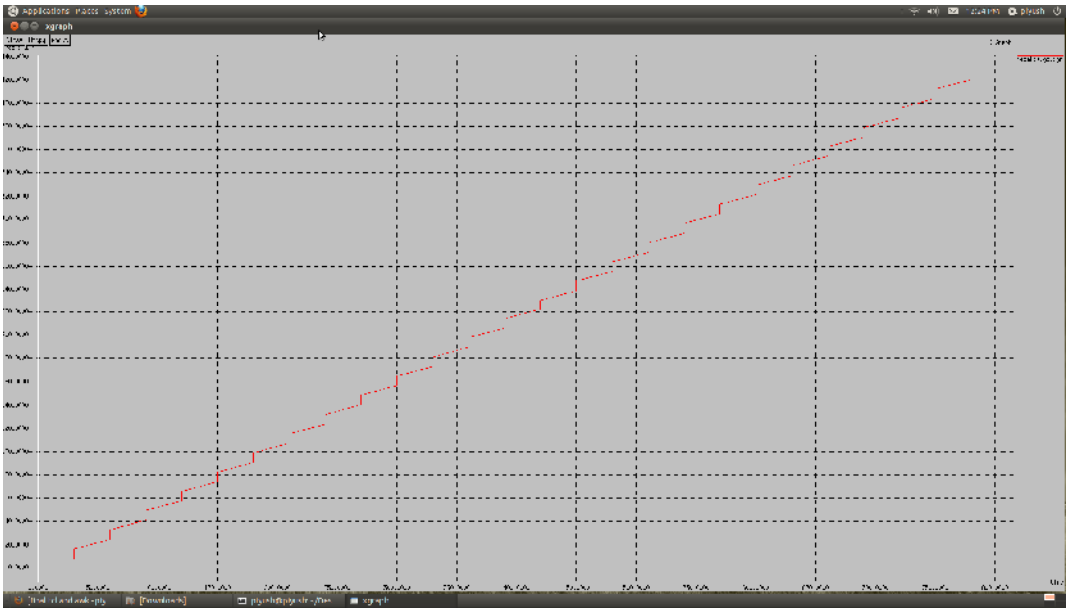
4.2.1 Throughput of sending packets

As seen from the graph that there is a steady increase of traffic for sending request to each other. Since the NchooseK algorithm is applied in this simulation
In both the cases there's a steady increase of traffic for sending request to each other. The total time taken for simulation is 150 seconds in which the data is transmitted and received among the nodes. Due to large number of keys made by nchoosek which have more number of common keys. This scheme tries to overcome from the imitations of PSK and the sending packet delivery ratio exist in favour of proper connectivity in WSN.



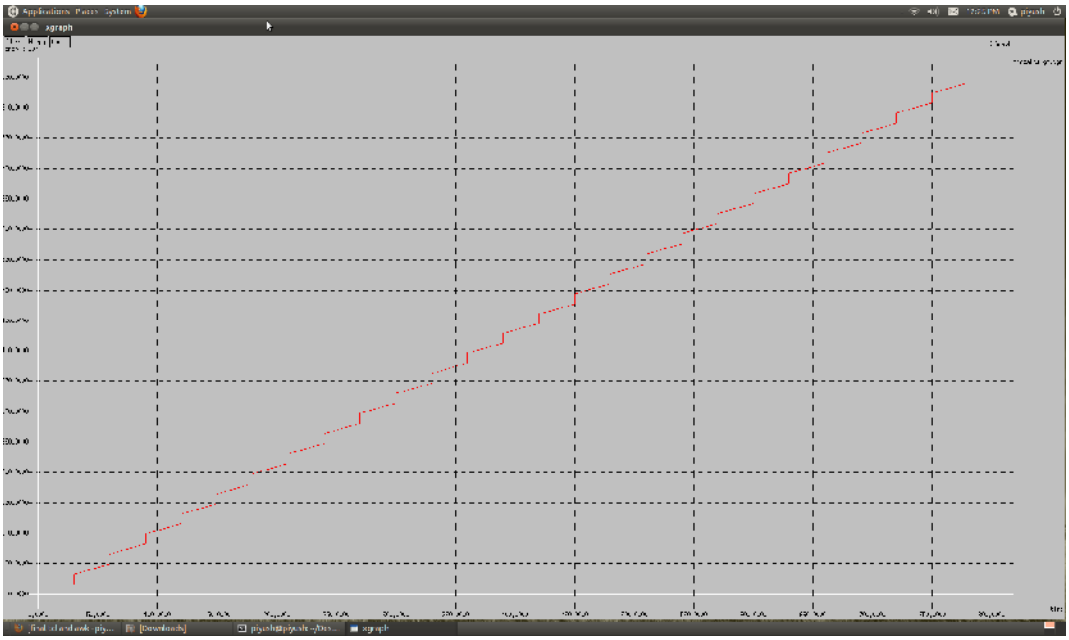
4.2.2Throughput of receiving packets

After implementing the algorithm NChooseK for improving the drawbacks of Key Pre-distribution scheme, the following graph is obtained as the packets are being received. The rate of received packets is increasing with the passage of time and there is a small number of packets dropped as compared to the receive before graph**.** The zig-zag line shows the effect of NChooseK algorithm. The receiver of packet delivery ratio in the above graph illustrates to overcome the limitations of resilience. It requires minimal cost, battery and power requirement which is far better as compared to the PSK scheme.
The graph illustrates that the resilience problem is also improved and in favour of resource efficient with connectivity and adaptability.



4.2.3 Drop

This graph depicts the number of packets which are dropped in the simulation by applying the NChooseK algorithm. As the number of packets received have increased substantially after applying the algorithm, so there is a decrease in the number of packets dropped. As there is less packet drop due to proposed scheme so there is less storage overhead , computational overhead and communication overhead. The graph also explains the high level of connectivity power in WSN. It also indicates the random key management for conducting the survey of the scheme which improves the effect of disjoint sets and much more in favour of WSN connectivity consumption.



5. CONCLUSION AND FUTURE WORK

The proposed scheme of Random key distribution in the wireless sensor network using the NchooseK algorithm has enhanced the packet delivery ratio and significantly reduced the drop rate of packets. This has enhanced the performance by reducing the congestion over the network. When connectivity of a network is affected due to some key scheme, it also affects many other parameters including energy consumed by the sensors which is a very critical parameter in area of sensors. Therefore for future, we suggest that more research should be done in every aspect. We have already covered the energy parameters namely send, receive and drop.

REFERENCES

[1] Laurent Eschenauer and Virgil D. Gligor. "A key-management scheme for distributed sensor networks." In CCS '02: Proceedings of the 9th ACM conference on Computer and communications security, pages 41- 47, New York, NY, USA, 2002. ACM.

[2] Ashok Kumar Das, " A Location-Adaptive Key Establishment Scheme for Large-Scale Distributed Wireless Sensor Networks", Journal Of Computers, Vol. 4, No. 9, September 2009.

[3] Chan H, Perrig A, Song D. "Random key pre-distribution schemes for sensor networks". In Proceedings of IEEE Symposium on Security and Privacy (S&P), 2003.

[4] Ji Heon Kwon, "Improved Connectivity Using Hybrid Uni/Omni-directional Antennas In Sensor Networks", Department of Electrical and Computer Engineering Texas A&M University.

[5] Noor J. Ottallah, "Implementation of Secure Key Management Techniques in Wireless Sensor Networks" , B.S University of New Orleans, May, 2008.

[6] Chi‐Yuan Chen, et.al., "A survey of key distribution in wireless sensor networks", Security Comm. Networks (2011) Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.354

[7] Zhang W, Tran M, Zhu S, Cao G. "A random perturbation‐based scheme for pair wise key establishment in sensor networks." In Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), September 9–14, Montreal, QC, Canada, 2007.

[8]http://www.codehappy.net/cspage/nchoosek.html , "NchooseK" .

[9] Sencun Zhu, et.al ," Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach.", Center for Secure Information Systems, George Mason University, Fairfax, VA 22030

[10] Subhankar Chattopadhyay, et.al., "Key Pre-distribution and Key Revocation in Wireless Sensor Networks", Department of Computer Science and Engineering National Institute of Technology Rourkela, Orissa, 769 008, India May 2011.