



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Pairwise Key Distribution Mechanism for Heterogeneous Sensor Network

Kanwalinderjit Gagneja
Department of Computer Science
Southern Oregon University
Ashland, Oregon, USA

Abstract: Generally the sensor nodes are deployed in hostile environment. Therefore, it becomes particularly significant that sensor nodes communicate data in a secure fashion. Since sensor nodes have limited resources, thus it is recommended by many researchers to use heterogeneous sensor nodes to increase the performance of the laid sensor network. In this paper, an efficient pairwise key distribution mechanism for heterogeneous sensor networks has been presented. The performance evaluation of the given mechanism shows optimal usage of sensor resources. The given mechanism assures the node connectivity in the network and each communication is made secure using cryptography. The nodes that get compromised do not disturb the functioning of rest of sensor network. However, it does affect the immediate neighbors of the compromised nodes. Moreover, if the network detects that a node is compromised the security breach could be effortlessly clogged. The security analysis of pairwise key distribution protocol (PKDP) shows that it is performing better than probabilistic key distribution techniques such as E-G while storing lesser number of keys and the communications are encrypted in a manner that increases its efficiency.

Keywords: Security; key distribution; pairwise; heterogeneous sensor networks

I. Introduction and Related Work

Nowadays, the use of sensor nodes has increased tremendously in our lives. In the coming ten to fifteen years these sensor nodes may possibly cover the world completely, and these tiny, little nodes could be accessed over the Internet. At present, these sensor nodes are used in various applications, for instance: traffic monitoring, nuclear power plant controlling and monitoring, global positioning system, environment monitoring, reporting of fire incidents, and vehicular movement tracking, etc. [3, 8, 9, 17]. These tiny sensor nodes have abundant possible uses for numerous applications, likewise: showbiz and entertainment industry, drugs and human trafficking control, border surveillance, sports industry, smart spaces, crisis management, drones, under water surveillance and environment monitoring, and health and medical industries [12]. Since all these applications handle sensitive data, so the sensor nodes should communicate data securely and efficiently [7].

It is well known among wireless sensor networks researchers that sensor nodes run with limited resources. The sensor nodes have limited processing power, have low bandwidth availability, have small transmitting range, and have inadequate memory [1, 15, 16]. Thus it becomes very significant to realize key arrangement in sensor networks because the nodes in sensor networks are resource constrained and communication security is mandatory. Usually key-distribution techniques hone on probabilistic key distribution. It could be seen in Eschenauer and Gligor (E-G) [5] random key management technique. The E-G technique works in three phases: The first phase is for key pre-distribution; in second phase it computes shared keys; in third and final step it sets up path keys. The technique given by E-G utilizes a considerable size of memory simply to store the keys. However, probabilistic key distribution techniques suffer from various adverse effects that public key based techniques do not.

The probabilistic key distribution techniques cannot assure that a given sensor node could establish a shared secret key with its neighbour sensor nodes. The probabilistic key distribution techniques also cannot assure security for the genuine and uncompromised sensor nodes, when there are a number of compromised sensor nodes in the given network. In [19], Zhu et al proposed Localized Encryption and Authentication Protocol (LEAP). It is again a probabilistic key distribution scheme. Main idea behind this protocol is that all the different communications happening among the sensor nodes need various security measures. Moreover, to satisfy different security needs it is not appropriate to use same security techniques. In [14] a new security protocol named Self-Configuring Bootstrapping Keys (SBK) is proposed, where the protocol does not require keys to be stored over the nodes before the network is deployed. The wireless sensor network supports two dissimilar kinds of sensor nodes: the service sensor nodes and the worker sensor nodes. The service nodes are responsible to generate key spaces and pass on the keys to worker sensor nodes. The worker sensor nodes then use these keys to bootstrap and establish pairwise keys with their neighbour sensor nodes.

On the other hand, it is shown in the research that Elliptical Curve Cryptography is useful for tiny sensor nodes [10]. Various current researches have presented that Heterogeneous Sensor Networks (HSNs) are good structural lay out to improve the sensor networks performance significantly [8, 13]. In the given HSNs there are

two dissimilar kinds of nodes: the high end sensor nodes and the low end sensor nodes. The low end sensor nodes are small, simple with little processing power and cheap, whereas the high end sensor nodes have good processing power, more memory, and are costly. Since the sensor nodes those are high end are better equipped with resources than low end sensor nodes, thus could be used to design significantly good security techniques.

The paper organization is as follows: Section II elaborates on the layout of the network and provides detail of key distribution, some assumptions, what different keys are stored on low end sensor nodes, and what different keys are stored on high end sensor nodes. Section III describes how one hop, two hops, and n hop clusters are generated by the high end sensor nodes, and how sensor nodes setup their keys. Section IV discusses the performance evaluation of the presented scheme in terms of memory usage, communication cost, and computation costs. Section V explains the weaknesses, comparison with probabilistic key distribution technique, and network connectivity. Section VI concludes the paper.

II. The Layout of Network with Key Distribution

This section presents the layout of the network and the key distribution technique. In the given area of interest thousands of low end sensor nodes are laid out and there could be hundreds of high end sensor nodes. The complete network is supported by one base station. All the nodes sense the data and forward it to the base station. The eavesdropper nodes usually have more sophisticated capabilities than the wireless sensor nodes in the laid network. The eavesdropper sensor nodes could be equipped with global overhearing capabilities, thus could compromise some low end wireless sensor nodes in the network. Unambiguously, some of the eavesdropper nodes capabilities are higher energies, more computation capability, and larger memories. In the following we present some assumptions those are followed before laying out the sensor nodes in the network.

- 1) In the wireless sensor network, every node is identified with its unique ID.
- 2) The high end sensor nodes are powerful nodes and are resistant to tempering, so high end sensor nodes cannot be compromised.
- 3) Every high end sensor node stores a common public key pair.
- 4) Since low end nodes are less capable so are not tamper resistant and invader or eavesdropper could capture them and these nodes could be compromised.

A. Key Management

The base station assigns the unique ID's to all the nodes and also pre-calculates the keys as well. There are two key sets one for low end sensor nodes and one for high end sensor nodes. Each low end sensor node is loaded with the key sets represented in table I.

Table I: Key sets used by low end nodes

Notation	Explanation
ID_a	An unique ID for low end node
K_a^R	Private Key
K_a^P	Public Key
$K_{i,BS}^P$	A public key shared between given low end node i and the only one base station
$[K_a^P ID_a]K_{BS}^R$	It is the certificate generated with the concatenation of public key and the unique ID, and it is encrypted with, K_{BS}^R , the private key of the base station

The keys could also be computed based on [7], where they have used an algorithm to compute the keys based on the node IDs and the level of that node in the given network. By doing that the nodes can self-certify the communication and the node does not have to communicate back to the high end node or cluster head for end to end communication.

Similarly, each high end sensor node is loaded with the keys represented in table II.

Table II: key sets used by high end nodes

Notation	Explanation
ID_H	An unique ID for high end node
K_H^R	Private Key
K_H^P	Public Key
$K_{i,BS}^P$	A public key shared between given high end node i and the base station
$[K_H^P ID_H]K_{BS}^R$	It is the certificate generated with the concatenation of public key and the unique ID, and it is encrypted with the, K_{BS}^R , private key of the base station

III. Clustering and Key Distribution

Once the network is deployed, the high end sensor nodes start reaching to other high end sensor nodes and low end sensor nodes around it to form clusters by broadcasting HELLO messages. The HELLO message is composed of: Node's unique ID_H , and its public key K_H^P that is encrypted with K_{BS}^R : $HELLO[K_H^P || ID_H]K_{BS}^R$.

The HELLO messages are broadcasted repeatedly after a random delay of time, so that high end sensor node could reach to maximum number of sensor nodes within its transmission range.

On the other hand, the low end sensor nodes keep receiving every HELLO message for some predetermined time frame. Once that time is over the low end node picks the HELLO message it received with maximum signal strength. The low end sensor node replies back to that high end node with its certificate to authenticate itself. The high end sensor node then sets up a shared secret key with that low end sensor node as follows:

First, low end sensor node sends $[K_a^P \parallel ID_a]K_{BS}^R$ and high end sensor node sends $[K_H^P \parallel ID_H]K_{BS}^R$ to authenticate each other.

Second, they compute the shared secret key

$$Key_{H,a} = K_H^R \times H_1(K_a^P \oplus ID_a) + K_{iBS}^P \quad (1)$$

$$Key_{a,H} = K_a^R \times H_1(K_H^P \oplus ID_H) + K_{iBS}^P \quad (2)$$

Here H_1 is a hash function and $\oplus$ is an XOR operation. This computation should check if $Key_{H,a} = Key_{a,H}$ then low end and high end nodes both are sharing the same secret key and that is what is expected. Now both nodes save this secret key in their neighbor tables. Similarly, all other nodes in the network set up shared secret keys.

At this point, all the high end sensor nodes have setup one hop cluster around themselves. Now each high end node will generate one hop symmetric cluster key $Key_{H,Ci}$, and will use private key, K_H^R , to encrypt it: $[Key_{H,Ci}]K_H^R$ then will distribute this cluster key to all its one hop neighbors.

After forming one hop clusters, the nodes start looking for multi hop low end and or high end sensor nodes. In the following fig. 1(a) and fig. 1(b) there is one high end sensor node surrounded by several one hop low end sensor nodes. In fig. 1(a) there is one low end node that is receiving HELLO messages from high end node but cannot transmit back since it does not have enough transmission power. So this low end sensor node must choose appropriate single hop neighbor to form multi hop communication connection to the high end sensor node being its cluster head.

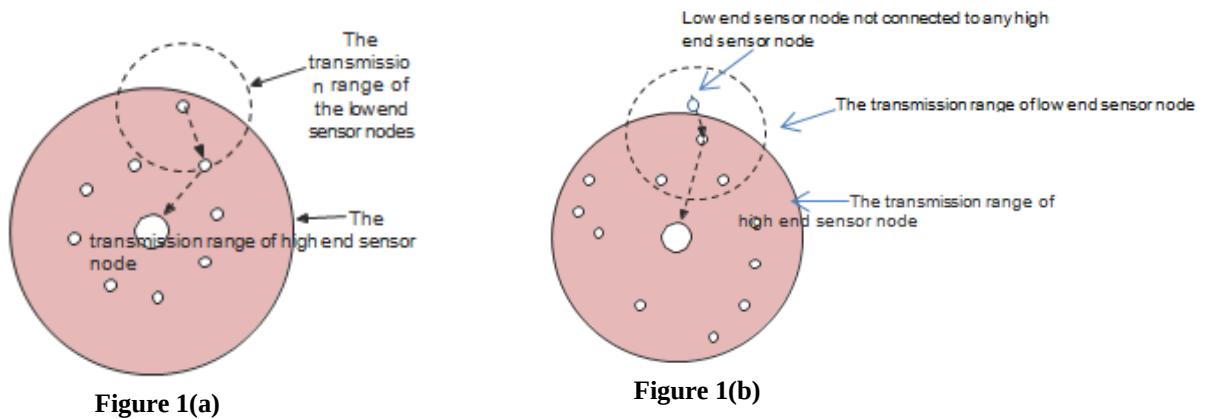


Figure 1(a): The multi hopped low end sensor node trying to reach to the high end sensor node, 1(b): One low end sensor node outside the transmission range of any high end sensor node, therefore not part of any cluster

In fig. 1(b) there is one low end sensor node that is not connected to any high end node. So this low end sensor node should pick another low end node within its transmission range so that it can establish pairwise communication with the high end sensor node. Using eq. (1) and (2) such nodes set up their pairwise keys for multi hop clusters. All the nodes in the network are given some time frame to set up pairwise keys with the neighbors within its radio range for secure communication of data.

The nodes are laid out in a manner that these form a connected graph. It means that each high end sensor node is within the transmission range of at least one other high end sensor node, so that these two nodes could send and receive signals from each other. The high end nodes also share secret keys as described in eq. (1) and (2) after authenticating each other. This will form high end sensor nodes cluster.

The base station also listens to the HELLO messages from the high end nodes and form cluster with those high end nodes with strongest transmission signals.

IV. Performance Evaluation

For the given pairwise key distribution protocol (PKDP) the performance is evaluated in terms of memory utilization, communication cost, and computational cost.

a) Memory Utilization

When the nodes are deployed in the given network, the nodes have some pre-deployed keys by the base station, for example, for high end sensor node it has: ID_H , K_H^R , K_H^P , K_{iBS}^P , one certificate, one group key. After the network deployment, the nodes set up pairwise keys with neighbors and cluster key with the other low end and high end sensor nodes to form the cluster. For high end sensor nodes it depends upon the degree, D_H , that is, the

number of low end and high end sensor nodes with which it can connect to establish the two way communication. Therefore memory requirements to store keys for high end sensor node is: the keys stored on the node before deployment plus the degree of high end node, D_H , and x bytes per key. So total memory used by high end sensor node is given by: $M_H = x \cdot (6 + D_H)$ bytes.

Similarly, the low end sensor nodes keep pre-deployed keys as: ID_a , K_a^R , K_a^P , $K_i^{P_{BS}}$, and one certificate. Say low end sensor node has a degree of, D_a , and it stores group keys for the neighbor connected to it. So total memory used by low end sensor node is given by: $M_a = x \cdot (5 + 2D_a)$ bytes.

Suppose we have one network, having 1000 low end nodes and 10 high end nodes in number. The average degree for high end nodes is $D_H=100$, and for low end nodes is $D_a=20$. Therefore, the total memory requirements by the complete network are given by $M = M_H + M_a$ bytes

$$M = 10x \cdot (6 + D_H) + 1000x \cdot (5 + 2D_a) \text{ bytes}$$

$M = 10x \cdot (6 + 100) + 1000x \cdot (5 + 2 \cdot 20)$ bytes or 46060 keys. If the node use 10 bytes to store one key, means $x = 10$ then total network memory requirement to store keys is 460600 bytes or less than 500 KB. Therefore each high end sensor node is storing 106 keys using 1060bytes or approximately 1KB of memory. Similarly, each low end node is storing 45 keys using 450 bytes or 0.44KB of memory. Presently, one TelosB mote possesses 10 KB RAM and 48 KB flash memory [2], so the given pairwise key distribution protocol (PKDP) could fairly easily implemented on such sensor nodes.

b) Communication Cost

The sensor nodes generally consume most of its energy while transmitting [6]. In the given pairwise key distribution protocol (PKDP) the network topology formation is quite efficient and the nodes perform the transmissions only two times as explained in section III. One when high end sensor nodes send HELLO messages to all its neighboring nodes, another when low end sensor nodes supply their certificate for authentication. Then both nodes compute their shared secret keys and if these two keys are similar then nodes are assumed to be authentic. The nodes then generate the group keys and the network can start communicating the data.

c) Computation Cost

In the given pairwise key distribution protocol (PKDP) the most computation intensive step is when high end nodes form clusters with other low end and high end sensor nodes. In cluster formation step the high end and low end sensor nodes compute their shared secret keys. Initially high end sensor nodes send HELLO messages and low end nodes authenticate themselves to high end node. After authentication they compute the shared secret key:

$$Key_{H,a} = K_H^R \times H_1(K_a^P \oplus ID_a) + K_i^{P_{BS}}$$

This operation is performed with one XOR operation, one hash function, one multiplication and one addition operation.

V. Security Analysis

It is made sure that in pairwise key distribution protocol (PKDP) every essential information is kept confidential. This way an eavesdropper cannot manage to break into the system. Once the nodes are deployment in the given network the low end and high end sensor nodes do not know their neighbors. The nodes do not know about which sensor nodes is transmitting to them with strongest transmission range. Moreover, the HELLO messages those are sent by high end sensor nodes are all encrypted $HELLO[K_H^P || ID_H]K_{BS}^R$ with the private key, K_{BS}^R . Even all the communications after the hello message are all encrypted. Thus makes this communication highly secure that an eavesdropper can neither read nor modify the contents of the communication.

a) Weaknesses

There could be some situation like: the enemy could capture the sensor node physically. If the captured node is high end sensor node then it is not much of a concern in terms of security for the given network. Since the high end sensor nodes are tamper resistant. Though, the network will not be able to gather any information from that cluster. In contrast, if the captured node is a low end node then for sure the attacker could extract the stored keys and could use those to collect more information from the network. But to collect information the hostile node has to use the certificate $[K_a^P || ID_a]K_{BS}^R$ to get data from other nodes. The most important key of all is $K_i^{P_{BS}}$. Other than base station it is mathematically impossible for other nodes to generate another certificate for authentication, thus base station will get to know about it. Once base station has the information about damaged or captured node, it will invalidate the node and will send new public shared key. The base station broadcasts the message to all other nodes in the network $\langle \text{FFFF}, [\text{INVALIDATE} || ID_i || ID_n || ID_p], K_{BS}^R \rangle$. Thus, other nodes do not send any data through invalidated node and start using new public shared key.

Sometime eavesdropper could intercept some message and could replay that message. Certainly, if this message replay happens during cluster formation, the nodes receiving this replay would not be able to understand this message. This replayed message could confuse them and could result into lost sensor node energy. This problem could be solved if we add sequence number to the messages. Hence, add a seq_n to the HELLO message that a high end node broadcasts $HELLO[K_H^P || ID_H]K_{BS}^R || [seq_n] K_H^R$. The receiving node will decrypt the message using $K_i^{P_{BS}}$ key and will verify the high end node. Then it will authenticate the sequence number. Only the

authentic high end node keeps the key, K_H^R , thus mathematically it is impossible that eavesdropper could fake the given sequence number.

a) *Comparison with Probabilistic Key Distribution techniques*

The E-G technique [5] gives the probability of connectivity for two host as in eq. (3):

$$p = 1 - (((P-k)!)/(P-2k)!P!) \quad (3)$$

The P refers to key pool size, and k represents the size of keychain [4]. Let key pool is 100 and the key chain is 5 then there is probability, p , of 23% that any two sensor nodes will have single shared key from the key pool, P . It signifies that that E-G just provides one fourth of connectivity chances for the 100 key pools. On the other hand with our technique having 106 keys, it shows that there is probability of 92% that any two nodes will share a key. Moreover, pairwise key distribution protocol (PKDP) uses approximately one third of the memory only.

With probabilistic key distribution, if the key pool, P , is large, it increases the security, instead decreases the chances that any two probable sensor nodes will have shared keys. However, with pairwise key distribution protocol (PKDP), the keys are computed cryptographically, so all keys are dissimilar except the, K_{iBS}^P , key and group keys.

b) *Connectivity*

For a complete network while deploying the nodes there is one concern of how many sensor nodes must be distributed so that the network is connected. This is represented in fig. 2, where probability to make a connected network is computed with respect to key pool size and key chain size and the partial data of complete computation using eq. (3) is presented in table III. From table III we can infer that if the key pool size is 3000 and key chain size is 121 then it is 99% probable that two sensors in a cluster will share a key. However, if the key pool size is 3000 and key chain size is 60 (almost half of the previously chosen key chain size) then it is approximately 71% probable that two sensors in a cluster will share a key. This means the sensor nodes are far apart. If the sensor nodes are far apart then they will be using longer transmission signals and may die soon as could be seen with probability2. However, to increase the network life time it is good idea that sensor nodes use shorter transmission range. This could be gained by increasing the small fraction of sensor nodes in the given network this could be seen with probability1 as presented in fig 2.

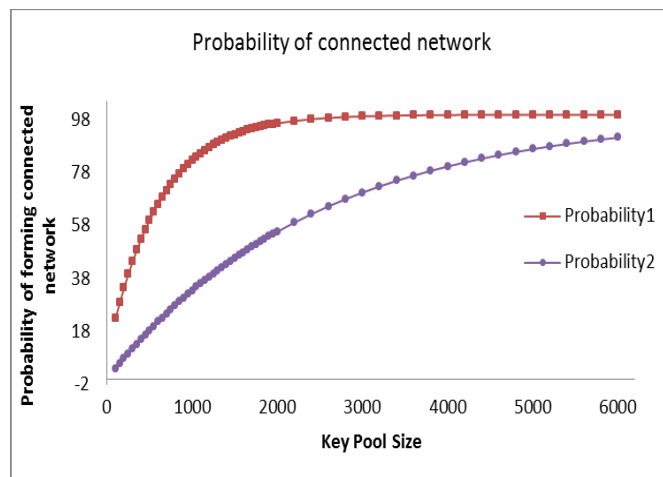


Figure 2: Two different probabilities of forming connected network

Table III Probabilities of making connected network

Key pool size	Key chain1	Prob1	Key chain2	Prob2
100	5	23	2	4
1000	41	82.67	20	33.5
1500	61	92.47	30	45.78
2000	81	96.73	40	55.79
3000	121	99.38	60	70.61
4000	161	99.88	80	80.46
5000	201	99.97	100	87
6000	241	99.99	120	91.36

This problem could also be solved by adding extra powerful transmitters, but we will just add extra sensor nodes for the connectivity. In other words we can state that:

Theorem 1: Finding the probability that a random undirected graph $G(x,y)$ is connected.

Proof: Say $f(x)$ is the probability of $G(x,y)$ to be connected and is given by

$$f(x) = 1 - \sum_{i=1}^{x-1} f(i) \binom{x-1}{i-1} (1-p)^{i(x-i)}$$

Now it is sufficient to show that

$$\sum_{i=1}^x f(i) \binom{x-1}{i-1} (1-p)^{i(x-i)} = 1$$

Since

$$\binom{x-1}{i-1} f(x) (1-p)^{x(x-i)} = f(x)$$

Suppose $v_1, v_2, v_3, \dots, v_n$ be n vertices.

Trivially,

$$1 = \sum_{i=1}^x P(v_i)$$

But

$$P(v_i) = \binom{x-1}{i-1} P(G(i,p) \text{ is connected}) (1-p)^{i(x-i)}$$

If A is set of all $i-1$ subset of $\{v_2, v_3, \dots, v_n\}$, then

$$\begin{aligned} P(v_i) &= \sum_{A \in \mathcal{A}} P(\text{the component of } v_i \text{ is } \{v_i\} \cup A) \\ &= \sum_{A \in \mathcal{A}} P(\{v_i\} \cup A \text{ is conn'd}) P(\text{no edge btwn } A \cup \{v_i\} \text{ \& } (A \cup \{v_i\})^c) \end{aligned}$$

The last statement tells that edges in $\{v_i\} \cup A$ are not dependent on the edges from A to A^c

In A there are only $\binom{x-1}{i-1}$ elements

The probability that $\{v_i\} \cup A$ is connected is equal to the probability that $G(1+|A|, p)$ is connected is $f(x)$.

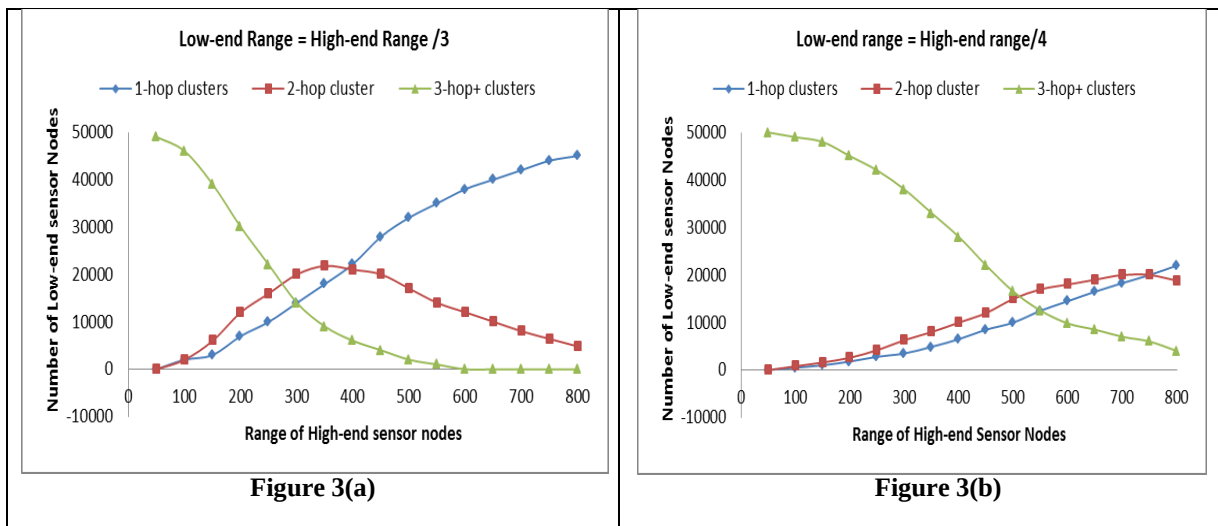


Figure 3(a): 1, 2, 3 hop+ clusters when low end sensor node range is one third that of high end nodes 3(b): 1, 2, 3 hop+ clusters when low end sensor node range is one fourth that of high end nodes

The low end sensor nodes are making either one hop or two hops or n hop clusters with its high end sensor node. Simulations are used to find the number of low end sensor nodes in one hop or two hops or n hop clusters and the results are presented in fig. 3(a) and fig. 3(b). In the presented data we have varied the high end sensor node range, and set the low end sensor nodes range equal to one third (as presented in fig. 3(a)) and one fourth (as presented in fig. 3(b)) of the high end sensor node. From fig. 3(a) and fig. 3(b) it could be observed that one hop clusters form the best configuration in terms of energy efficiency otherwise both the plots are almost similar. Another observed point is when high end sensor nodes have high transmission range then there is fairly good number of low end sensor nodes in one hop clusters than in two hop or n hop clusters.

VI. Conclusion

This paper presents pairwise key distribution protocol (PKDP) for heterogeneous sensor network. The performance evaluation of the given mechanism shows optimal usage of sensor resources, such as little memory usage, fewer computations, and smaller communication costs. The given mechanism assures the connectivity of the network and each communication is made secure using cryptography. The given technique makes sure that compromised node does not disturb the functioning of rest of sensor network. However, it does affect the immediate neighbors of the compromised nodes. Moreover, if the network detects that a node is compromised the security breach could be effortlessly clogged. The security analysis of pairwise key distribution protocol (PKDP) shows that it is performing better than probabilistic key distribution techniques such as E-G while storing lesser number of keys and the communication mechanism is encrypted in a manner that increases its efficiency.

VII. References

- [1] Blundo, Carlo, Santis, Alfredo De, Herzberg, Amir, Kuten, Shay, Vaccaro, Ugo, Yung, Moti, "Perfectly-Secure Key Distribution for Dynamic Conferences", 12th International Cryptology Conference on Advances in Cryptology, Santa Barbara, CA, pages 471-482, 1992.
- [2] Crossbow Technology Inc. URL: <http://www.xbow.com>.
- [3] Du, Wenliang, Deng, Jing, Han, Yunghsiang S., Varshney, Pramod, Katz, Jonathan, and Khalili, Aram, "A Pairwise Key Predistribution Scheme for Wireless Sensor Networks," Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), 2003.
- [4] Duarte-Melo E. J., and Liu M., "Data-gathering wireless sensor networks: Organization and capacity," Computer Networks, vol. 43, no. 4, pp. 519-537, 2003.
- [5] Eschenauer, Laurent and Gligor, Virgil D. "A Key- Management Scheme for Distributed Sensor Networks," Proc. Ninth ACM Conference on Computer and Communication Security (CCS '02), pages 41-47, November 18-22, 2002.
- [6] Heinzelman, Wendi, Rabiner, Chandrakasan, Anantha, and Balakrishnan, Hari, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," HICSS, Jan 4-7, 2000.
- [7] Gagneja, Kanwal, "Pairwise Key Distribution Scheme for Two-Tier Sensor Networks", IEEE ICNC, Honolulu, Hawaii, USA, pages 1081-1086, Feb. 3-6, 2014.
- [8] Gagneja K.K. and Nygard K., "Energy Efficient Approach with Integrated Key Management Scheme for Wireless Sensor Networks", ACM MOBIHOC, Bangalore, India, pages 13-18, July 29, 2013.
- [9] Gagneja K.K. and Nygard K., "Heuristic Clustering with Secured Routing in Heterogeneous Sensor Networks", IEEE SECON, New Orleans, USA, pages 9-16, June 24-26, 2013.
- [10] Gura N., Patel A., Wander A., Eberle H., and Shantz S.C., "Comparing elliptic curve cryptography and rsa on 8-bit cpus," in Proc. of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, Massachusetts, Aug. 2004.
- [11] Intanagonwiwat, Chalermek, Govindan, Ramesh, and Estrin, Deborah, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks," Proceedings of ACM Mobicom'00, pages 56-67, Aug 2000.
- [12] Karlof, Chris, and Wagner, David, "Secure Routing in Sensor Networks: Attacks and Countermeasures," Ad Hoc Networks, Vol. 1, Issue 2-3, pages 293-315, Sept 2003.
- [13] Liu, Donggang, Ning, Peng, Li, Rongfang, "Establishing Pairwise Keys in Distributed Sensor Networks", Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS '03), pages 52-61, October 2003.
- [14] Liu, Fang, Cheng, Xiuzhen, Ma, Liran, and Xing, Kai, "SBK: A Self-Configuring Framework for Bootstrapping Keys in Sensor Networks," IEEE Transactions on Mobile Computing, Vol. 7, Issue 7, pages 858-868, July 2008.
- [15] Li Qun, Rus, Daniela, "Global Clock Synchronization in Sensor Networks", IEEE Transactions on Computers, pages 214-226, 2004.
- [16] Madria, Sanjay, and Yin, Jian, "SeRWA: A Secure Routing Protocol Against Wormhole Attacks in Sensor Networks," Ad Hoc Networks, Vol. 7, Issue 6, pages 1051-1063, Aug 2009.
- [17] Perrig, Adrian, Szewczyk, Robert, Tygar J. D., Victorwen, and Culler, David E., "SPINS: Security Protocols for Sensor Networks," Journal of Wireless Networks, Vol. 8, Issue 5, pages 521-534, Sept 2002.
- [18] Pallapa G.V., "A privacy enhanced situation-aware middleware framework for ubiquitous computing environments", dissertation PhD, University of Texas, Arlington, 2010. <http://hdl.handle.net/10106/4867>
- [19] Roland M., Langer J., and Scharinger J., "Security vulnerabilities of the ndef signature record type," 3rd International Workshop on Near Field Communication (NFC), pages 65-70, Feb. 2011.
- [20] Yang, Yi, Wang, Xinran, Zhu, Sencun, and Cao, Guohong, "SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks," ACM Transaction on Information and System Security, Vol. 11, Issue 4, pages 1-43, July 2008.
- [21] Zhu, Sencun, Setia, Sanjeev, and Jajodia Sushil, "LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," ACM Transactions on Sensor Networks, Vol. 2, pages 500-528, Nov. 2006.

VIII. Acknowledgments

This research is partially funded by ETIC and by E-Tech. We thank all involved in the process of supporting this research.