Significant Storage on Sensor Storage Space, Energy Consumption and Better Security Based on Routing in Hybrid Sensor Networks

K.Nageswara rao[#1], Dr. D. Rajya Lakshmi[#2], Prof. T. Venkateswara rao[#3]

[#1]Research Scholar, GITAM University, Vishakhapatnam
ksn_choudary@yahoo.com
[#2]Department of Information Technology, GITAM University, Vishakhapatnam
rdavuluri@yahoo.com
[#3]Department of Computer Science Engineering, KL University
tv_venkat@yahoo.com

Abstract

WSNs are characterized by limited resources in terms of communication, computation and energy supply. A critical constraint on sensors networks is that sensor nodes employ batteries. A second constraint is that sensors will be deployed unattended and in large numbers, so that it will be difficult to change or recharge batteries in the sensors .The Energy Consumption in wireless sensor networks varies greatly based on the protocols the sensors use and computations used to generate keys for communication among neighbor nodes. Previous research on sensor network security mainly considers homogeneous sensor networks, where all sensor nodes have the same capabilities. Research has shown that homogeneous ad hoc networks have poor performance and scalability. The many-to-one traffic pattern dominates in sensor networks, and hence a sensor may only communicate with a small portion of its neighbors. Key Management is a fundamental security operation. Most existing key management schemes try to establish shared keys for all pairs of neighbor sensors, no matter whether these nodes communicate with each other or not, and this causes large overhead and more energy consumption and more storage requirement. In this paper, we adopt a Hybrid Sensor Network (HSN) model for better performance and security. We propose a novel routing-driven key establishment scheme, which only establishes shared keys for neighbor sensors that communicate with each other. We utilize Elliptic Curve Cryptography in the design of an efficient key Establishment scheme for sensor nodes. The performance evaluation and security analysis show that our key Establishment scheme can provide better security with significant reductions on communication overhead, storage space and energy consumption than other key Establishment schemes.

General terms

This is my research paper related to wireless sensor networks.

Key words:

storage space, energy consumption, Security, key usage, sensor network, Elliptic Curve Cryptography.

DOI : 10.5121/ijasuc.2011.2402

1. INTRODUCTION

A *Sensor Network* is a wireless, ad hoc network, made of a large number (hundreds or thousands) of nodes, whose positions occur randomly .Sensor networks have applications in many areas, such as military, homeland security, health care, environment, agriculture, manufacturing, and so on. A critical constraint on sensors networks is that sensor nodes employ batteries. A second constraint is that sensors will be deployed unattended and in large numbers, so that it will be difficult to change or recharge batteries in the sensors. Therefore, all systems, processes and communication protocols for sensors and sensor networks must minimize power consumption.
A general definition of a *sensor* is “a device that produces measurable response to a change in a physical or chemical condition”, more specifically, a sensor is "a device that responds to a stimulus, such as heat, light, or pressure, and generates a signal that can be measured or interpreted". The Sensor Network community often (but not always) defines a sensor node as a small, wireless device, capable of responding to one or several stimuli, processing the data and transmitting the information over a short distance using a radio link. Sensor nodes employ electronic circuits that minimize power consumption. Typically sensors are thought of as measuring light, sound and temperature. However, sensors can measure other variables, such as electromagnetic fields or vibrations. Sensor transmits values wirelessly to one or several sinks. Most previous work on sensor networks considered homogeneous sensor networks, i.e., all sensor nodes have the same capability in terms of communication, computation, energy supply, storage space, reliability. A homogeneous ad hoc network has poor fundamental limits and performance. Research has demonstrated its performance bottleneck both theoretically [1, 2] and through simulation experiments and tested measurements [3]. In this paper we use heterogeneous nodes in sensor networks. Recently deployed sensor network systems are increasingly following heterogeneous designs, incorporating a mixture of sensors with widely.

Security is critical to sensor networks deployed in hostile environments, such as military battlefield. Security issues in homogeneous sensor networks have been extensively studied. Key management is an essential cryptographic primitive upon which other security primitives are built. Several key management schemes have been proposed for homogeneous sensor networks. In [9], Eschenauer and Gligor first present a key probabilistic pre-distribution scheme for key management in sensor networks. Later, a few other key pre-distribution schemes (e.g., [10-13]) have been proposed. Probabilistic key pre-distribution is a promising scheme for key management in sensor networks. To ensure the scheme works well, the probability that each sensor has at least one shared key with a neighbor sensor (referred to as key-sharing probability) should be high. For the key pre-distribution scheme in [9], each sensor randomly selects its key ring from a key pool of size P. When the key pool size is large, each sensor needs to pre-load a large number of keys to achieve a high key-sharing probability. For example, when P is 10,000, each sensor needs to pre-load more than 150 keys for a key-sharing probability of 0.9 [9]. If the key length is 256 bits, then 150 keys require a storage space of 4,800 bytes. Such a storage requirement is too large for many sensor nodes. For example, a smart dust sensor [14] has only 8K bytes of program memory and 512 bytes of data memory.

The above discussion shows that many existing key Establishment schemes (e.g., [9-13]) require a large storage space for key pre-distribution and are not suitable for small sensor nodes. In this paper, we present an efficient key Establishment scheme that only requires small storage space of sensor nodes. The scheme achieves significant storage saving varying capabilities [4]. For example, a sensor network may include small MICA sensors as well as more powerful high -end nodes such as robotic nodes [4]. Several recent literatures [5-8] have studied non-security aspects of HSN. However, security issues of HSN remain largely unexplored by utilizing an efficient public key algorithm and the fact that a sensor node only communicates with a small portion of its neighbors.

Most existing key Establishment schemes for sensor networks are designed to set up shared keys for all pairs of neighbor sensors, without considering the actual communication pattern. In many sensor networks, sensor nodes are densely deployed in the field. One sensor could have as many as 30 or more neighbors [15]. The many-to-one traffic pattern dominates in typical sensor networks, where all sensors send data to one (or a few) sink. Because of the many-to-one traffic pattern, a sensor node only communicates with a small portion of its neighbors, e.g., neighbor sensors that are in the routes from itself to the sink. This means that a sensor node does not need shared keys with all neighbors. Below we give a definition that considers the fact.

ECC can be combined with Diffie-Hellman approach to provide key exchange scheme for two communication parties. ECC can also be utilized for generating digital signature, data encryption and decryption. The Elliptic Curve Digital Signature Algorithm (ECDSA) utilizes ECC to generate digital signature for authentication and other security purposes [19, 20]. Several approaches for encryption and decryption using ECC have been proposed [16, 17, 19].In this paper, we present an efficient key Establishment scheme for HSN which utilizes the c-neighbor concept and ECC public-key cryptography. Typical sensor nodes are unreliable devices and may fail overtime. Our key management scheme considers communication topology change caused by node failures, i.e., the scheme set up pair wise keys for each sensor with more than one neighbor. In case the primary next-hop node fails, a backup node is used for communications. In addition, if there is a need for two neighbor sensor nodes to set up shared keys later (e.g., in case all backup nodes fail); they can do this with the help from other neighbors [9]. The rest of the paper is organized as follows. Section II presents the key Establishment scheme based on routing. Section III gives Methodologies Section IV gives the Implementing Algorithms and Results security analysis. Section IV concludes this paper.

2. METHODOLOGY

2.1. Key Establishment Scheme based on Routing

In this Section, we present an efficient key Establishment scheme for HSN which utilizes ECC and the many-to-one communication pattern in sensor networks. The scheme is referred to as ECC-based key management scheme. We adopt a realistic model of HSN that can be used in most sensor network applications. The HSN model consists of a small number of powerful high-end sensors (H-sensors) and a large number of low-end sensors (L-sensors). Both H-sensors and L-sensors are powered by batteries and have limited energy supply. L-sensors use multi-hop communications to reach H-sensors, and H-sensors use multi-hop communications to reach the sink.

2.1.1. Key maintenance

In First, we list the assumptions of HSN below.

1. Due to cost constraints, L-sensors are not equipped with tamper-resistant hardware. Assume

that if an adversary compromises a L-sensor, she can extract all key material, data, and code stored in that node.

2. H-sensors are equipped with tamper-resistant hardware. It is reasonable to assume that powerful H-sensors are equipped with the technology. In addition, the number of H-sensors in a HSN is small (e.g., 20 H-sensors and 1,000 L-sensors in a HSN). Hence, the total cost of tamper-resistant hardware in a HSN is low.

3. Each L-sensor (and H-sensor) is static and aware of its own location. Sensor nodes may use secure location services such as [23] to estimate their locations, and no GPS receiver is required at each node.

4. Each L-sensor (and H-sensor) has a unique node ID.

5. The sink is well protected and trusted.

Since H-sensors are powerful nodes, key establishment for H-sensors are relatively easy. For example, each H-sensor can be pre-loaded with a special key K_H , which is protected by the tamper resistant hardware. After deployment, two H- sensors can use K_H to achieve secure communications. In this paper, we focus on key establishment for L-sensors.

The notations used in the rest of the paper are listed below:

- *u, v, x, y, n* are L-sensors; H is a H-sensor;
- $\{m\}_k$ denotes encrypting message *m* with key *k*.

Next, we briefly describe cluster formation in HSN.

2.2. Tree-Based Cluster Formation in HSN

We adopt a typical assumption of sensors' locations as Some both L-sensors and H-sensors are uniformly and randomly distributed in the network. The tree based cluster given as follows from that the siblings (L-sensors) are neighbors of the H-sensor. The communication in between these two sensors is the data will pass from H-sensor.the initial configure is called as intracluster. This clustering designing described clearly in [0]. Note that our key management scheme does not rely on such sensor distribution, i.e., it also works well for other sensor distributions. After sensor deployment, clusters are formed in a HSN. We have designed an efficient clustering scheme for HSN in [22] also. Because of the page limit, we will not describe the details of the clustering scheme in this paper. For the simplicity of discussion, we assume that each H-sensor can communicate directly with its neighbor H-sensors (if not, then relay via L-sensors can be used). All H-sensors form a backbone in a HSN. After cluster formation, a HSN is divided into multiple clusters, Where H-sensors

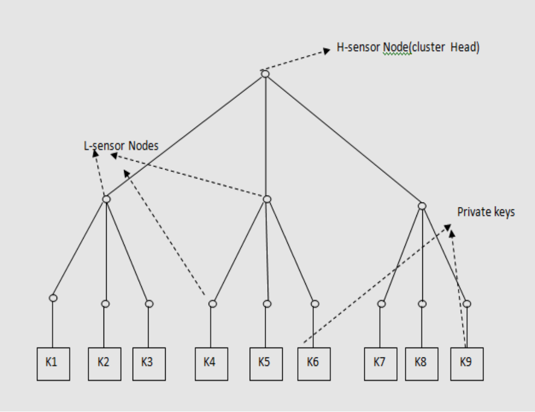


Fig: 1 Tree based Cluster Formation in HSN

Serve as the cluster heads. If the tree is depicts as two-dimensional plane for each each sub tree, each L-sensor selects the closest H-sensor as the cluster head. And from the above tree structure each sub tree can be treated as inter clusters and the starting generation is intracluster.

H-sensor--------→cluster head.
L-sensor-------→sibling of that parent (but not all, if each sub tree consider as one cluster).
K1, k2----------kn-----→shared private keys (L-sensors having these keys).

2.2.1 .communication of sensors in HSN

In a HSN, the sink, H-sensors and L-sensors form hierarchical network architecture. Clusters are formed in the network and H-sensors serve as cluster heads. All H-sensors form a communication backbone in the network. Powerful H-sensors have sufficient energy supply, long transmission range, high date rate, and thus provide many advantages for designing more efficient routing protocols. We have designed an efficient routing protocol for HSN in [23]. Routing in HSN consists of two phases: 1) Intra -cluster routing: Each L-sensor sends data to its cluster head (a H-sensor); and 2) Inter-cluster routing: Each cluster head may aggregate data from multiple L-sensors and then sends compressed data to the sink via the H-sensor backbone. The routing structure in HSN is illustrated in Figure 1. Before discuss key establishment for L-sensors, we briefly describe the intra-cluster routing scheme in [23].

2.3. ECC- based key management scheme

Due to page limitation I am not giving complete description. In [0] and I was given a detailed explanation in ECC-based key management scheme subsection.

3. AIM AND CALICULATIONS

In this section, we evaluate that minimization of storage space, power consumption and status of security and also comparisons in EG and ECC schemes in further subsections.

3.1. Storage saving

Storage can be reduced in sensor networks by reducing number of keys to be stored in each sensor node. in a cluster, assume H-sensor=M and L-sensor=N. each L-sensor is preloaded with its private key and public key of H-sensor. Each H-sensor is preload with public keys of all L-sensors, a pair of private and public key for itself and a key K_H for newly deployed sensors.

E-G scheme

Preloaded keys= p *(No of L –sensors+ No of H-sensors)

ECC Scheme

Preloaded keys= (No of H –sensors+2)* No of L-sensors+3*No of H-sensors

3.1.1. Algorithm

```
Algorithm Storage _saving ()
{
        //Enter no of nodes in a Sensor Network;
        int n;
        //Identify L-sensors and H-sensors
        int L.H;
        // Form the Cluster with 1 H-sensor and m L-
            Sensors
        Cluster_ form (H.m);
        //Cluster Head Registration
        H-sensor nodeID= ClusterHead_registration (node H)
        // Register all L-sensors in a cluster, with cluster head
        for i=1 to m
            NodeID=Node_ registration (node I, location);
        //Generate preloaded keys to H-sensor and H-sensor using Elliptic Curve Cryptography
(ECC);
        ECC ();
        //Preload the keys in H-sensor and L-sensor;
        Preload _keys (H, m);
}
ClusterHead_registration (node H)
{
        Return nodeID;
}
Node registration (Node L, location)
{
        Return nodeID;
}
ECC ()
{
        Choose E (a, b) with an Elliptic curve over GF (p);
        Choose a point on the curve say e1(x1, y1);
        Choose an integer d;
        Calculate e2(x2, y2) =d*e1(x1, y1);
        Return e1, e2 and E (a, b) as a public keys and‘d’ as private keys
}
```

3.1.2. Calculations for Storage Saving

Formulae

E-G scheme

Preloaded keys= p *(No of L –sensors+ No of H-sensors)

ECC Scheme

Preloaded keys= (No of H –sensors+2)* No of L-sensors+3*No of H-sensors

Consider

i) L-sensors=200, H-sensors=20,p=80

In E-G scheme

Preloaded keys= 80*(200+20)=17600

In ECC scheme

Preloaded keys= (20+2)*200+3*20=446

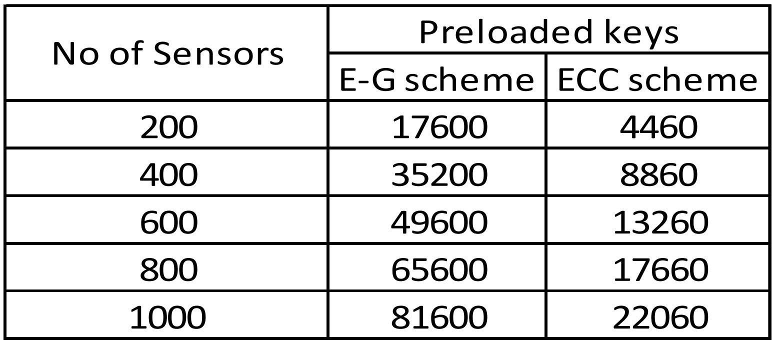
ii) L-sensors=400, H-sensors=20, p=80

In E-G scheme

Preloaded keys= 80*(400+20) =35200

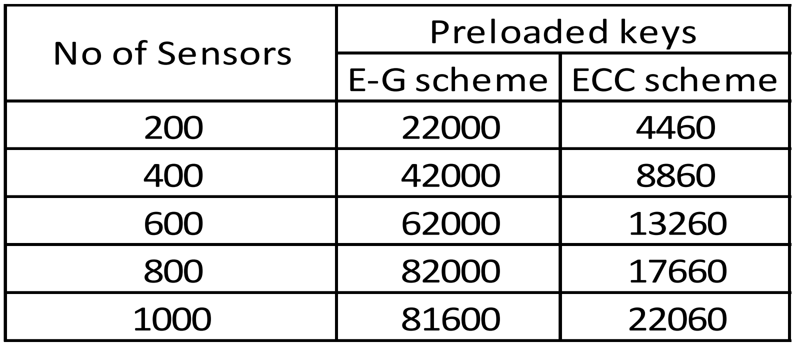
In ECC scheme

Preloaded keys=(20+2)*400+3*20=8860



No of Sensors	Preloaded keys	
	E-G scheme	ECC scheme
200	17600	4460
400	35200	8860
600	49600	13260
800	65600	17660
1000	81600	22060

Fig 2: Table with p=80



No of Sensors	Preloaded keys	
	E-G scheme	ECC scheme
200	22000	4460
400	42000	8860
600	62000	13260
800	82000	17660
1000	81600	22060

Fig 3: Table with p=100

3.2. Energy Consumption

The Energy Consumption is calculated based on routing in HSN. Routing in Sensor Networks can be inter-cluster and intra-cluster routing. An intra-cluster routing scheme determines how to route packets from a L-sensor to its cluster head. When a L-sensor sends a packet to its cluster head (say H), the packet is forwarded by other L-sensors in the cluster. We use Figure 1 to describe an intra-cluster routing scheme. The basic idea is to let all L-sensors (in a cluster) form a tree rooted at the cluster head H. It has been shown in [22] that: (1) If complete data fusion is conducted at intermediate nodes, (i.e., two *k*-bit packets come in, and one *k*-bit packet goes out after data fusion) then a minimum spanning tree (MST) consumes the least total energy in the cluster. (2) If there is no data fusion within the cluster, then a shortest-path tree (SPT) consumes the least total energy. (3) For partial fusion, it is a NP-complete problem of finding the tree that consumes the least total energy.

For sensor networks where data generated by neighbor sensors are highly correlated (e.g., two *k*-bit packets are aggregated to one *m*-bit packet, where *m* is close to *k*), a MST may be used to approximate the least energy consumption case. To construct a MST, each L-sensor sends its location information to the cluster head H, and then H can run a centralized MST algorithm to construct the tree. After constructing the MST, H can disseminate the tree structure (parent-child relationships) to all L- sensors using one or more broadcasts. For example, a pair (*u*, *v*) can be used to denote that L-sensor *u* is *v*'s parent node. If the cluster is small, one broadcast message can include all the pairs. If the cluster is large, then it can be divided into several sections. The H can notify L-sensors in each section by one broadcast. Note that the broadcast from a cluster head needs to be authenticated. Otherwise, an adversary may broadcast malicious messages and disrupt the dissemination of routing information. We discuss the broadcast authentication in next subsection. For sensor networks where the data from neighbor sensors have little correlation, a SPT can be constructed; using either centralized or distributed algorithms.

Since L-sensors are small, unreliable devices and may fail overtime, robust and self-healing routing protocols are critical to ensure reliable communications among L-sensors. During the tree setup, the MST or SPT algorithm can find more than one parent nodes for each L-sensor. One parent node serves as the primary parent, and other parent nodes serve as backup parents. In case the primary parent node fails, a L-sensor uses a backup parent for routing.

Given the tree- based routing structure within a cluster, each L-sensor only needs to establish shared keys with its r-neighbors, i.e., its parent-nodes and child-nodes.

Energy consumption is reduced if use reduces number of transmissions and receiving of keys. In ECC-scheme we establish keys (shared keys) with communication neighbors.

3.2.1. Algorithm

```
Algorithm Energy_consumption ()

{
        //Construct MST using centralized MST algorithm;
 MST_construct (H, m);
        // Key Establishment for each L-sensor to communicate with their neighbors
        For i=1 to m
          $K_{I, v}$ =Key_request (H-SensorID, L-sensorID, location)
        //Communication with their neighbor sensors with shared keys
        Communicate (u, v, Ku, v)
}
MST_construct (H, m)
{
        Send location of L-sensor to H;
        Use Centralized MST algorithm to construct tree;
        H broadcast message to all sensor about parent-child relationship;
        Two or more parent nodes are determined for each L-sensor. One serves as Primary
        parent and other serve as backup parents.
}
Key_request (node H, node L, Location)
{
        H generates a shared key and sends it to L-sensor;
}
```

3.2.2. Calculations for Energy Consumption

We calculate the energy consumption for Establishing shared keys for communication only, not for data transmissions.

Assume, the energy

For Transmission (E_{tx}) = 81mW
For Receiving (E_{rx}) =32mW
For idle (E_{id}) =12mW

Consider no of sensor n=200
L-Sensors=200
H-Sensors=20
n=2(communication neighbors)

In ECC Scheme

Clusters are formed with 11 sensors (10 L-sensors, 1 H-sensor) for each cluster

Energy consumption for shared key establishment= energy consumption for MSTconstruct and send the keys to the respective sensors

For one cluster

H-sensor receives a Message from all L-sensors

Energy consumption = no of L-sensors*(E_{tx} + E_{rx})

E1=10*(81+32) =1130

H-Sensor Broadcasts tree structure to all L-sensors

Energy consumption = no of L-sensors*(E_{tx} + E_{rx})

E2=10*(81+32) =1130

H generates shared keys for each L-sensor and its r-neighbors

For 1 sensor = 3* 81+3*32+7*12=243+96+84=423

For 10 sensor=10*423=4230

Total Energy Consumption for one cluster= 4230+E1+E2=4230+1130+1130=6490mW

For 20 clusters =20 * Energy for 1 cluster=20*6490=**129800mW**

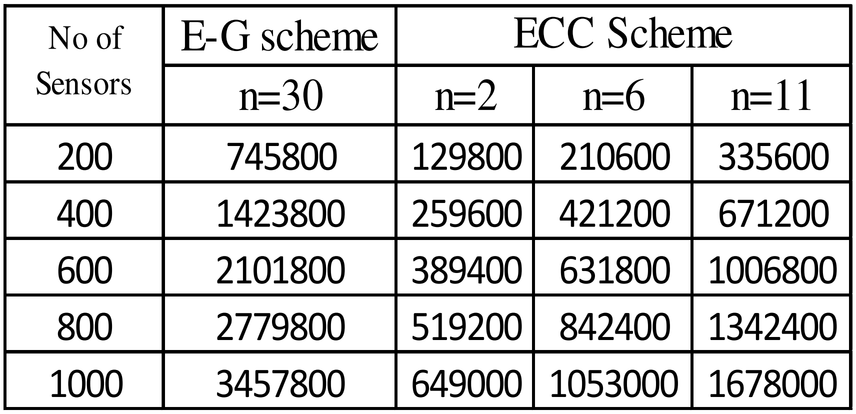
In E-G Scheme

Consider the neighbor sensors are 30

We need to generate 30 pair-wise keys for each sensor

Therefore, the Energy consumption for one sensor= 30*(81+32) =3390mw

For 200 sensors (20 additional sensors are considered) = 3390*20=**745800mW**



No of Sensors	E-G scheme	ECC Scheme		
	n=30	n=2	n=6	n=11
200	745800	129800	210600	335600
400	1423800	259600	421200	671200
600	2101800	389400	631800	1006800
800	2779800	519200	842400	1342400
1000	3457800	649000	1053000	1678000

Fig 4: Table with n=2, 6, 11

4. Result Analysis With Graphs.

4.1 Comparison of Storage Saving with E-G and ECC scheme

In Wireless sensor networks, the storage space is limited so we need to reduce the size of the keys stored on sensor nodes .Here; we compare the keys generated in E-G scheme and ECC scheme (which uses Elliptic Curve Cryptography).

p=80 *10^3

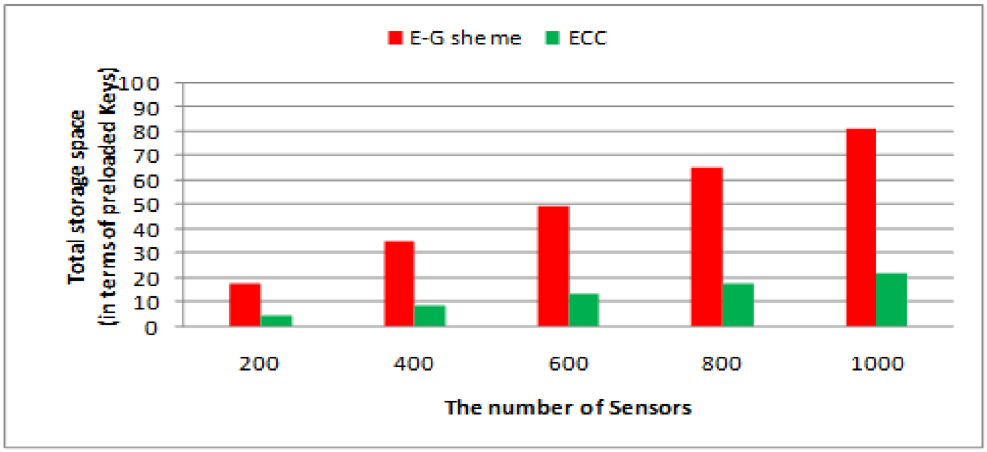


Fig 5: Storage saving p=80

p=100 *10^3

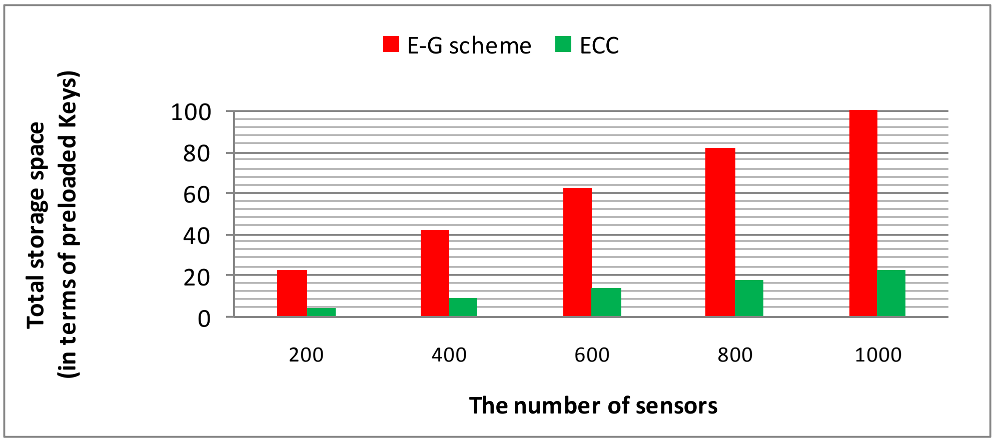


Fig 6: Storage saving p=100

Here P is no of loaded keys to achieve a Key sharing probability high in E-G scheme, where as there is no need of loaded keys in ECC scheme. P value depends on Key Pool size, if key pool size is high, the p becomes high. To achieve Key sharing probability high in E-G scheme the Key Pool size is 10000 and loaded keys are 150 then the probability is 0.9

4.2. Comparison of Energy Consumption with E-G and ECC Scheme

Here we consider only the energy consumption for establishing shared keys not for data transmission

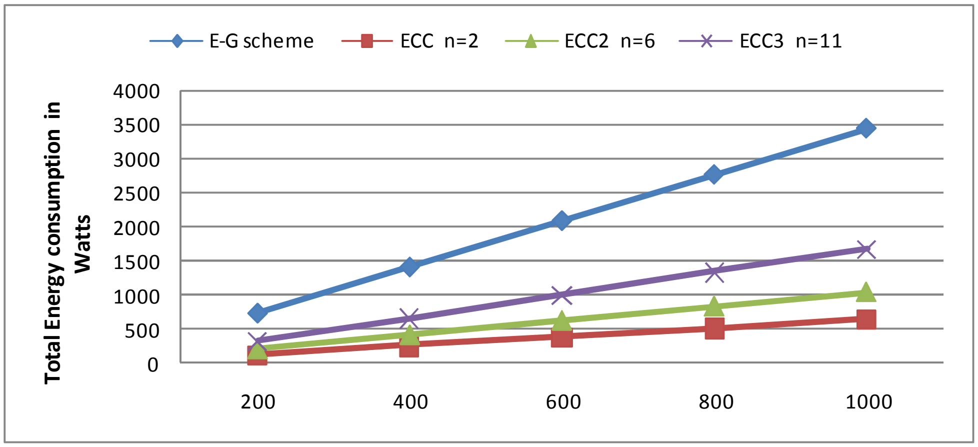


Fig 7: Storage saving p=80

5. SECURITY ANALYSIS

In this subsection, we analyze the resilience of our ECC-based key Establishment scheme against node compromise attack. We want to find out the effect of *c* L-sensors being compromised on the rest of the network, i.e., for any two L-sensors *u* and *v* which are not compromised, what is the probability that the adversary can decrypt the communications between *u* and *v* when *c* L-sensors are compromised? The probability is referred to as the *compromising probability*.

In the ECC-based scheme, each L-sensor is pre-loaded with one unique private key. After key setup, each pair of Communicating L-sensors has a different shared key. Thus, compromising *c* L-sensors does not affect the security of communications among other L-sensors.

In [10], Chan *et al.* calculate the probability that two sensors have exactly *j* common keys in the E-G scheme:

$$P(i) = \binom{p}{j}\binom{p-j}{2(m-j)}\binom{2(m-j)}{m-j} / \binom{p}{m}^2$$

Where *m* is the number of pre-loaded keys in each sensor. Chan *et al.* compute the *compromising probability* under the E-G scheme as:

$$\mathrm{C(m)} = \sum_{k=0}^{n}\left(1-\left(1-\frac{m}{p}\right)^c\right)^j p(j) / \sum_{j=1}^{m} p(j)$$

In Figure 8, we compare the *compromising probability* under the ECC-based scheme and the E-G scheme. The number of compromised sensors – *c* varies from 10 to 200, with an increment of 10. For the E-G scheme, the key pool size P is 10,000, and we calculate the *compromising* *probability* for three different values of *m* (the number of pre-loaded keys in each sensor): 20, 30, and 50. Figure 8 shows that the larger the *m*, the larger the *compromising probability*, i.e., less resilient to node compromise attack. For the ECC-based scheme, the compromising probability is

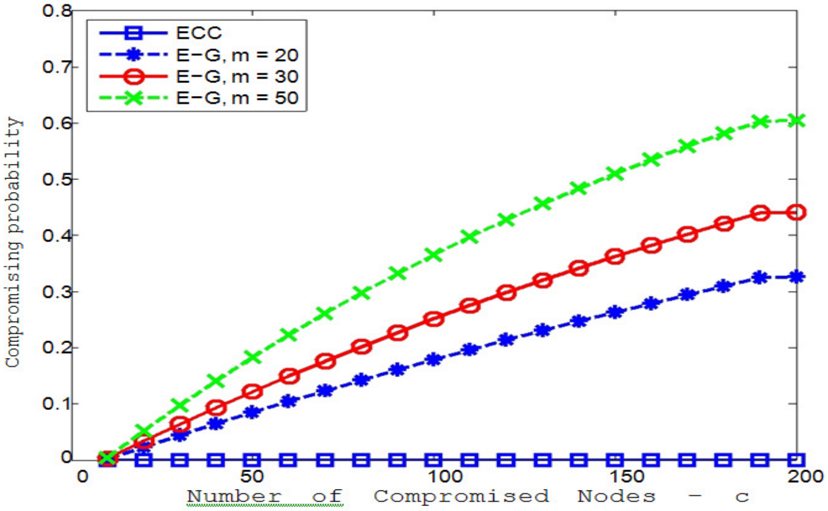


Fig 8: Security comparison

Always zero, no matter how many sensors are compromised. Thus, the ECC-based key management scheme has high resilience against node compromise attack.

6. CONCLUSION

In this paper, I proposed a efficient key management scheme than several existing systems for hybrid sensor networks. In our sensor networks we efficiently reduces the sensor storage space, power consumption and achieved better security (i.e. in the node compromise attack node is behave as fine resilience). For achieve this presented an ECC-based cryptography scheme. In this scheme the fact utilization of sensors in network is within a small portion of its neighbors is greatly reduces communication and computation overheads of key setup.

7. ACKNOWLEDGEMENT:

I very much thankful to all authors of their valuable reference papers for preparing this research paper and also thank full to my guides for their valuable guidance.

8. REFERENCES

[1] K.Nageswara Rao, L.V.Krishnarao, M. Ramakrishnam Raju, Dr D.Rajya Lakshmi "An Ad-hoc Key establishment based on routing for Heterogeneous Sensor Networks", international journal of advanced engineering sciences and technologies vol no,4 issue no,1,pp 015-020 Aug.2011.

[2] P. Gupta and P. R. Kumar, "The Capacity of Wireless Networks," IEEE Trans. on Information Theory, vol. IT-46, no. 2, pp. 388-404, Mar. 2000.

[3] E. J. Duarte-Melo and M. Liu, “Data-gathering wireless sensor networks: organization and capacity,” Computer Networks, Vol. 43, Issue 4, pp. 519-537, Nov. 2003.

[4] K. Xu, X. Hong, M. Gerla, “An Ad Hoc Network with Mobile Backbones,” Proc. of IEEE ICC 2002, New York, NY, Apr. 2002.

[5] L. Girod, T. Stathopoulos, N. Ramanathan, et al., “A System for Simulation, Emulation, and Deployment of Heterogeneous Sensor Networks,” Proc. of ACM SenSys 2004.

[6] S. Rhee, D. Seetharam, and S. Liu, “Techniques for Minimizing Power Consumption in Low Data-Rate Wireless Sensor Networks,” Proc. of IEEE WCNC’04, Atlanta, GA, March, 2004.

[7] R. Cristescu, and B. Beferull-Lozano, “Lossy Network Correlated Data Gathering with High-Resolution Coding,” Proc. of IEEE IPSN 2005.

[8] H. Wang, D. Estrin, and L. Girod, “Preprocessing in a Tiered Sensor network for Habitat Monitoring,” Proc. of IEEE Conf. on Acoustics, Speech, and Signal Processing, Hong Kong, China, April 2003.

[9] M. Yarvis, N. Kushalnagar, H. Singh, et al., “Exploiting Heterogeneity in Sensor Networks,” Proc. of the IEEE INFOCOM, Mar. 2005.

[10] L. Eschenauer and V. D. Gligor, “A key management scheme for distributed sensor networks,” Proc. of the 9th ACM CCS, Nov. 2002.

[11] H. Chan, A. Perrig, D. Song, “Random Key Predistribution Schemes for Sensor Networks,” Proc. of the 2003 IEEE Symposium on Security and Privacy, May 11-14, 197 – 213.

[12] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” Proc. of the 10th ACM CCS, pp 42-51, Washington D.C., Oct., 2003.

[13] S. Zhu, S. Setia and S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks,” Proc. of the 10th ACM CCS, Washington D.C., Oct., 2003.

[14] W. Du, J. Deng, Y.S. Han. P. K. Varshney, “A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks,” Proc. of the 10th ACM CCS, pp 42--51, Washington D.C., Oct., 2003.

[15] J. M. Kahn, R. H. Katz and K. S. J. Pister, “Mobile Networking for Smart Dust,” Proc. of ACM MobiCom, Seattle, WA, Aug., 1999.

[16] K. Whitehouse, C. Sharp, E. Brewer, D. Culler, “Hood: a Neighborhood Abstraction for Sensor Networks,” Proc. of ACM MobiSys'04, Boston, MA, June, 2004.

[17] N. Koblitz, “Elliptic curve cryptosystems,” Mathematics of Computation 48, 1987, pp203–209.

[18] V. Miller, “Use of elliptic curves in cryptography,” CRYPTO 85, 1985.

[19] N. Gura, A. Patel, A. Wander, H. Eberle, and S. C. Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs,” Proc. of the 6th International Workshop on Cryptographic Hardware and Embedded Systems, Boston, MA, Aug. 2004.

[20] N. Koblitz, “A Course in Number Theory and Cryptography,” Second Edition, Graduate Texts in Mathematics, Vol. 114, Springer, 1994.

[21] I. Blake, G. Seroussi, N. Smart, “Elliptic Curves in Cryptography,” London Mathematical Society, Lecture Note Series 265, Cambridge University Press, 1999.

[22] L. Lazos and R. Poovendran, “SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks”, in Proc. of 2004 ACM workshop on Wireless security (ACM WiSe 2004), Philadelphia, PA.

[23] X. Du and F. Lin, “Maintaining Differentiated Coverage in Heterogeneous Sensor Networks,” EURASIP Journal on Wireless Communications and Networking, Issue 4, pp 565–572, 2005.

[24] X. Du and Y. Xiao, “Energy Efficient Chessboard Clustering and Routing in Heterogeneous Sensor Network,” International Journal of Wireless and Mobile Computing (IJWMC), to appear.

[25] R. Cristescu, and B. Beferull-Lozano, “Lossy Network Correlated Data Gathering with High-Resolution Coding,” Proc. of IEEE IPSN 2005.

[26] B. Karp and H. T. Kung, “Gpsr: greedy perimeter stateless routing for wireless networks,” Proc. of the 6th ACM MobiCom, pp. 243-254, 2000.

[27] F. Kuhn, R. Wattenhofer, and A. Zollinger, “Worst-Case Optimal and Average-Case Efficient Geometric Ad-Hoc Routing,” Proc. of the 4th ACM MobiHoc, 2003.

[28] QualNet simulator, Scalable Network Inc., www.qualnet.com.

[29] MICA2 Mote Datasheet, www.xbow.com.

Authors

K.Nageswara Rao is an Associate Professor and Head in the Department of Computer Science & Engineering, Mother Teresa Institute of Science and Technology, Sathupally, Khammam (Dt). Mr.Rao received M.Sc (Computer Science) from Bharatidasan University, Tiruchy, 2000, M.Tech (Computer Science & Engineering) from Bharath University, Chennai, 2005 and currently pursuing Ph.D (Computer Science & Engineering) - Part-Time from GITAM University.

Dr. D. Rajya Lakshmi is Professor & Head, in the Department of Information Technology, GIT, GITAM University, Vishakhapatnam. Dr. D. Rajya Lakshmi completed B.E (Electrical) Degree from Andhra University in 1992, M.Tech (Computer Science & Engineering) from Andhra University in 1995 and received Ph.D (Computer Science & Engineering) from JNTUH.

Dr. T.V. Rao is currently working as Professor in the department of Computer Science & Engineering in K L University, Vijayawada. Dr. Rao completed B.E (Electronics & Communication Engineering) from Andhra University, Vishakhapatnam in 1977, M.Tech (computer S cience & Engineering) from PSG College of Engineering, Coimbatore in 1979 and Ph.D (computer Science & Engineering) from Wayne State University,Detroit,USA,1992.