Improving Connectivity and Resilience using ECC based Key Predistribution for deployment knowledge based WSN

R.KISHORE, S.RADHA, L.CHERLYFLAR
Electronics and Communication Engineering Department
Sri Sivasubramaniya Nadar College of Engineering
Kalavakkam, Chennai-603110
INDIA
kishorer@ssn.edu.in radhas@ssn.edu.in cherlyflar@gmail.com

Abstract: - Wireless Sensor Networks consist of small nodes with sensing, computation and communication capabilities, often deployed in remote inaccessible areas to interact with the environment and forward the measured event to the base station, thus posing several possibilities of physical attacks. Hence security becomes an important issue in wireless sensor networks. For resource constrained environments like wireless sensor networks key predistribution is found to be better choice. Another significant information regarding network is the deployment knowledge which can be used to improve the performance of the key predistribution schemes. Deployment knowledge offers numerous advantages when used in resource constrained environments, achieving better storage, better resilience to node capture, minimizing the number of keys and reduce network overhead. The main objective of this paper is to propose a novel scheme where keys are generated using Elliptic Curve Cryptography and predistributed into the nodes. Nodes are deployed in the area of interest considering hexagonal deployment knowledge and the links are formed based on the common keys of private key ring in each node. The performance of the system is evaluated in terms of resilience and connectivity. The results show that the connectivity and resilience of the network is better when compared to other existing key predistribution schemes.

Key-Words: - Security, Key Predistribution, Elliptic Curve Cryptography, Elliptic Curve Points, Connectivity, Resilience

1 Introduction

A sensor network is comprised of large number of nodes which are deployed densely to measure an event. Each of these nodes collects data and its purpose is to route this information back to a sink. Wireless sensor networks are infrastructure less and can operate in any environment as compared to traditional networks. Wireless Sensor Networks are limited in their energy, computation and communication capabilities [1]. Sensor networks applications include environmental monitoring, health monitoring, habitat monitoring, intrusion detection, forest fire and volcanic eruption detection etc. When sensor networks are deployed in hostile environments security becomes more important as they are prone to different types of malicious attacks.

The security requirement is to provide confidentiality, integrity, authenticity, and availability of all messages in the presence of adversaries. In order to establish a secure communication among sensor nodes secure link has to be established using key agreement schemes. Number of key management schemes was proposed to improve the security in wireless sensor network. With the support of effective key management schemes, the information transmitted in the network can be protected from any external attacker. Key distribution mechanism must support large network, and must be flexible against substantial increase in the size of the network even after deployment.

Blom (1985) proposed a secure key predistribution scheme where each node stores relatively small secret and public data from which it can derive a unique pair wise key for any neighbor. Each node stores a private matrix and a column of a public matrix. To establish a pairwise key between nodes, the nodes first exchange their public column information and then each makes partial matrix multiplication with the private information. It has the property that, as long as no more than λ nodes are compromised, all communication links of non-compromised nodes remain secure [2]. This solution unfortunately involves complex vector

multiplication thereby making scalability a concern. The reason being the computation becomes more costly if the size of the network increases. Another important issue in this scheme is the resilience, because if an adversary discovers the private matrix, there is a possibility that the entire network is compromised.

The basic probabilistic key pre distribution scheme is introduced by Eschenauer and Gligor [3]. The scheme consists of three phases: key predistribution, shared key discovery, and path key establishment. In this scheme each node randomly picks up a set of keys from a large key pool. After the sensors are deployed into the field, each node tries to find a common key to establish a secure connection. For those nodes that don't have a shared key, a path key establishment through an intermediate node is formed. Eschenauer and Gligor ensures that only a small number of keys need to be placed on each sensor node's key ring to ensure that any two nodes share (at least) a key with a chosen probability; e.g., for a probability of 0.5, only 75 keys drawn out of a pool of 10,000 keys. If the pool size is 100,000 then the number of keys required is still only 250 [3]. Thus the basic scheme is a key management technique that is scalable, flexible and can also be used for large distributed sensor networks. Trade-offs in the basic scheme can be made between sensor memory and connectivity but, it does not provide the node to node authentication property that ascertains the identity of a node with which another node is communicating.

When the network size is large, predistribution of secret keys for all pairs of nodes is not viable due to the large amount of memory used. Du et al., proposed a Key Predistribution Scheme, which employs Blom's scheme for key generation [4]. This scheme exhibits a nice threshold property. The threshold λ can be treated as a security parameter. The selection of larger λ value leads to greater resilience. The threshold property of Blom's scheme is a desirable feature because one can set the value of λ such that an adversary needs to compromise a significant fraction of the network in order to achieve any payoff. When the number of compromised nodes is less than the threshold, the probability that any node other than these compromised nodes is affected is close to zero. However, increasing λ also increases the amount of memory required to store key information.

Another scheme proposed by Du et al [5] involves the deployment knowledge of the nodes. This scheme assumes a grid deployment mechanism. Nodes are assumed to be deployed in the center of each zone as a batch. Those batches of nodes are distributed over each zone according to Gaussian distribution, which is best fit to the real world deployment scenarios. In this deployment model, the nodes in each batch are assumed to be close to each other. Keys are assigned to each node randomly by selecting from the key pool of the corresponding zone. Each zone share keys with its neighbor key zones. In this way, the nodes that are close to each other have a probability to share keys, but the distant nodes do not. The advantage of the scheme is that prior deployment knowledge reduces unnecessary memory assignments. It is more resilient than Blom's scheme with the same amount of memory required.

DDHV scheme with deployment knowledge assumes grid deployment (Du et al., 2004). In this scheme, nodes are divided into groups and each group is given a subset of key pool. Since nodes pick keys from a smaller pool, less memory is needed to achieve higher connectivity [6]. If adversaries randomly compromise nodes among the entire network, the scheme is more resilient than the basic scheme as discussed earlier. However, it could not keep the same performance when adversaries compromise nodes within a small local area. Zhen Yu and Yong Guan (2005) [7] proposed a hexagonal grid based scheme in which the sensor field is divided into hexagonal grids and the nodes are divided into groups, each of which is deployed into a grid. The sensor field is divided into hexagonal grids. The center of grid is the deployment point, which is the desired location of a group of nodes. The location of sensor node over the entire sensor field follows some distribution with a probability density function. In most cases, sensor nodes are often assumed to be uniformly deployed, i.e., the uniform distribution.

In hexagon-based scheme, all adjacent sensor nodes have the same distance. The hexagon system has some advantages over the rectangular system. First, in transmitting data, the signal range is more appropriate than rectangular system. Second, the distance between the neighboring sensors nodes differ, depending on whether the neighboring node is located directly adjacent or diagonal to it. Based on the deployment model, a group of nodes are deployed in a small local area, which causes most neighbors of a node come from its own group or neighboring groups. In the hexagon based scheme, each sensor node takes its deployment hexagon as the center and share keys with the sensor nodes deployed in its 19 adjacent hexagons. This scheme also consists of three phases namely, key predistribution phase, shared key discovery phase and path key establishment phase as discussed

earlier. The scalability of the scheme is good with good resilience and higher connectivity [7]. The scheme has good storage with good communication and normal power rating. The drawback of the scheme is that the entire deployment depends on the accurate model of the locations. If this breaks down, the entire scheme breaks down.

Most of the key predistribution schemes in the literature provide higher connectivity with considerable amount of resilience or vice-versa. In the proposed work, the objective is to achieve higher connectivity as well as increased resilience with reduced memory requirement. Here the keys are generated using ECC, as it has shorter key size, reduced network overhead and same level of security when compared to other methods [8]. The shorter key size makes the storage effective as these nodes have smaller memory and lesser energy. The keys generated using ECC are called seed keys and each node in the network is assigned with a unique seed key with which they generate a private key ring. Then the private key rings are predistributed in to the sensor nodes which increase the resilience of the network and deployed into the field following hexagonal deployment knowledge as it reduces unnecessary key assignments with increased connectivity.

2 Proposed Scheme

In the proposed scheme keys are generated using point addition and point doubling elliptic curve arithmetic operations. Generated keys are called as seed keys. A unique seed key is assigned to each node with which they generate a private key ring. Finally the private key rings are predistributed into the sensor nodes prior deployment. The proposed scheme uses hexagonal grid as the deployment model which reduces unnecessary key assignments.

2.1 Mathematical modeling

ECC make use of elliptic curves in which the variables and coefficients are all restricted to elements of finite field [8]. Cryptographic applications use two families of curves such as prime curve and the binary curve. As the prime curves are best suited for software implementations, the work is based on prime curve where a cubic equation is used in which the variables and coefficients all take on values in the set of integers from 0 through p-1 and in which the calculations are performed using modulo p, where p is the prime number chosen based on the network requirements.

2.1.1 Elliptic curve over finite fields

The prime curves are defined over finite odd prime field F_p [9]. The elliptic curves over a finite field are defined as shown in (1). A point on the elliptic curve can be represented as $P=(x,y)$ where $x,y \in F_p$. The modulo p function performs a wrapping around operation so that the elements are all within F_p. Thus the geometric shape of the curve is not preserved whereas its abelian properties are intact. Fig.1 shows the points in the elliptic curve $E_{107}(9,17)$. It can be seen that there is no geometric interpretation of the curve over a finite field. However the points show symmetry, as the points above and below are additive inverses of each other, wrapped around by modulo p operation.

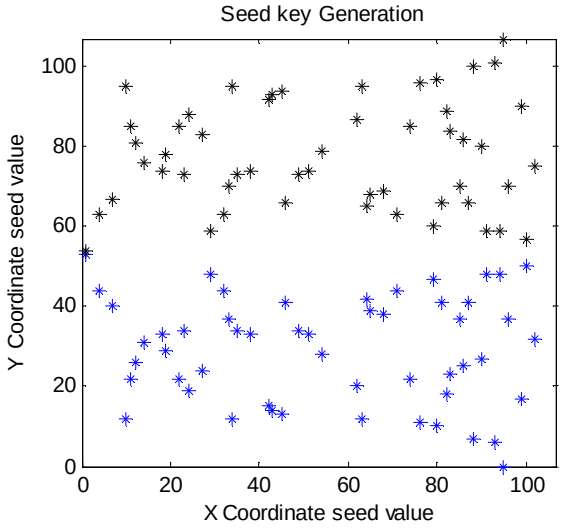

Fig.1. Points over elliptic curve $E_{107}(9, 17)$

2.1.2 Elliptic curve arithmetic operations

In this section the mathematical aspects that are involved in the generation of elliptic curve points are presented in detail. The basic elliptic curve operations are: Point addition, point doubling, point negation and point multiplication [10]. The general cubic equation for the elliptic curve is given by

$$y^2 \bmod p = x^3 + ax + b \bmod p \qquad (1)$$

where x, y are variables; a, b are coefficients. For given values of a, b, the points consists of positive and negative values of y for each value of x. Thus each curve is symmetrica1ly arranged. The chosen variables should satisfy the condition:

$$4a^3 + 27b^2 \bmod p \neq 0 \bmod p \qquad (2)$$

The abelian group can be defined on the basis of a set $E_p(a,b)$ where p is modular prime integer making the elliptic curve finite field. The number of points in a finite abelian group $E_p(a,b)$ is bounded by $p+1-2\sqrt{p} \le N \le p+1+2\sqrt{p}$.

Point Addition

Consider two different points $M=(x_m,y_m)$ and $N=(x_n,y_n)$ which are non negatives of each other, i,e $M \neq N$, the slope Δ is given by

$$\Delta = \frac{y_n - y_m}{x_n - x_m} \bmod p \quad (3)$$

To find the sum of the two points i.e., $M+N=Q$ the algebraic manipulation has to be done as shown below:

$$x_q = \Delta^2 - x_m - x_n \bmod p \quad (4)$$

$$y_q = -y_m + \Delta(x_m - x_q) \bmod p \quad (5)$$

When $y_m \neq 0$ the above expression can be written as

$$x_q = \left(\frac{3x_m^2 + a}{2y_m}\right)^2 - 2x_m \bmod p \quad (6)$$

$$y_q = \left(\frac{3x_m^2 + a}{2y_m}\right)(x_m - x_q) - y_m \bmod p \quad (7)$$

Point Doubling

If $M=N$ then $M+N=2M$ then point doubling equations are used. Consider a point M= (x_m,y_m), where $y_m \neq 0$ and $Q=2M$ where Q=(x_q,y_q) ,then the expressions for generating the elliptic curve points are represented as:

$$x_q = \Delta^2 - 2x_m \bmod p \quad (8)$$

$$y_q = -y_m + \Delta(x_m - x_q) \bmod p \quad (9)$$

where Δ is a tangent at point M, and it is given by

$$\Delta = \left(\frac{3x_m^2 + a}{2y_m}\right) \bmod p \text{ ,if } M=N \quad (10)$$

If $y_m = 0$ then $2M=O$, where O is the point at infinity.

Point Multiplication

Point multiplication operation is carried out through repeated addition. Consider a point M on the elliptic curve, which is multiplied with a scalar k using elliptic curve equation to obtain another point N. Point multiplication, can be achieved by point addition and point doubling operations. For example to find $N=kM$, if k=23, then

$$kM = 23.M = 2(2(2(2M)+M)+M)+M$$

This method is also known as 'double and add' method as it involves both point addition and point doubling operations to find the result of point multiplication.

2.2 Hexagonal deployment model

In practice the nodes are deployed in groups [8]. Each group of node is deployed around a single deployment point which is the desired location of the nodes. Such a group based deployment is assumed and the deployment knowledge is modelled. The sensor field is divided into t number of hexagonal grids equally. N numbers of nodes are divided into equal groups and are deployed into the hexagonal grids. The deployment can be modelled by a normal distribution and it is given by

$$f_i(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-[(x-\mu_{x_i})^2 + (y-\mu_{y_i})^2]}{2\sigma^2}} \quad (11)$$

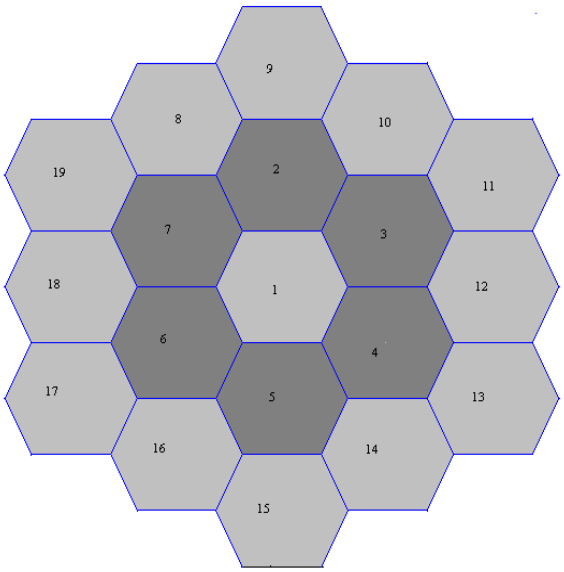

Fig.2. Hexagonal Grid

2.3 Key distribution

Key distribution involves four phases such as key generation phase, key predistribution phase, shared key discovery phase and path key establishment phase. The first two phases are done offline and remaining phases take place online.

2.3.1 Key generation phase

The seed points are generated according to the general equation of elliptic curve, the parameters *a*, *b* are chosen thereby satisfying (2) based on the network requirements. Large integer *p* should be chosen as a prime number, which is greater than the total number of nodes in the network. All the elliptic curve points generated lies in the finite field of the elliptic curve. The points are arranged in a symmetrical fashion. For the chosen elliptic curve group $E_p(a,b)$ the non negative integers in the quadrant from (0,0) through $(p-1, p-1)$ are alone interested. It can also be shown that a finite abelian group based on the set $E_p(a,b)$ provided that $x^3 + ax + b$ has no repeated factors. Each node is assigned with a unique seed key with which they generate private key rings.

2.3.2 Key predistribution phase

The key predistribution is done prior to deployment. The generated private key rings are predistributed into the nodes and deployed into the field. It is hard for the attacker to break this scheme as each seed point is unique.

2.3.3 Shared key discovery phase

Once the nodes are deployed into the field shared key discovery phase takes place. The key sharing between node 1 and node 2 is accomplished as follows:

1) Each and every node is assigned with an elliptic curve point as its seed key.
2) Based on the seed key, every sensor node in the network generates a key ring using point addition and point multiplication operation.

The idea behind the shared key discovery is when any two nodes in the network share a common key from the key ring then those two nodes establish a secured link.

2.3.4 Path key establishment phase

If any two nodes do not find a common key during the shared key discovery phase, then those nodes try to establish a secure connection through the intermediate nodes.

3 Modeling of attack

Security is always a critical issue in wireless sensor network. When an adversary physically captures one or more sensor nodes, all the information stored on these nodes may be exposed completely [11]. Consequently the attacker can use the captured information to compromise the remaining part of the network. So it is essential that the network security is maintained no matter how the adversary tries to attack [12]. A few among the attack patterns are described here.

3.1 Random attack model

Random attack is the most widely used attack pattern in wireless sensor network research. Here the probability of a malicious node compromising a legitimate node is assumed to be 0.5. Under such probability we are finding how many times the network is entirely compromised. Once the nodes are captured randomly, the attacker tries to decrease the resilience of the network by capturing the keys that are predistributed in the captured node. Thus the random attack poses a serious threat to the confidentiality of the information present in the network, thereby affecting the security.

Algorithm:

```
x coordinate value;
y coordinate value;
Initialize:
communication range;
threshold value;
total number of nodes;
total number of malicious nodes;
for i=1:number of nodes
      distance= sqrt((node(x)-malicious node(x))^2 +
                (node(y)-malicious node(y))^2)
if (distance < communication range)
if (probability of malicious node compromising
   a legitimate node = = 0.5)
    node capture through random function++
end
end
if(node capture through random function>threshold)
  ntw_capture through random function++
end
end
```

3.2 Closest Neighbor attack model

In the closest neighbor attack pattern, when an attacker finished compromising a sensor node, the adversary may try to find another sensor node which is closest neighboring node to the current compromised node. This attack pattern is realistic and occurs frequently in a practical scenario.

Algorithm:

```
x coordinate value;
y coordinate value;
Initialize:
communication range;
threshold value;
total number of nodes;
total number of malicious nodes;
for i=1:number of nodes
      distance= sqrt((node(x)-malicious node(x))^2 +
                (node(y)-malicious node(y))^2)
if (distance<communication range)
if(probability of malicious node compromising
  a legitimate node = = 0.5)
   node capture through closest neighbor function++
end
for i= 1:number of nodes
for j=2:number of  nodes
     distance1= sqrt((nodei(x)-malicious nodej(x))^2 +
                (nodei(y)-malicious nodej(y))^2)
 if(distance1<communication range)
   neighbor count++
 if(probability of malicious node compromising
   a legitimate node = = 0.5)
    node capture through closest neighbor
    function++
end
end
end
end
end
end
if(node capture through closest neighbor
  function> threshold)
   ntw_capture through closest neighbor function++
end
```

3.3 Sybil attack model

In Sybil attack, a single node presents multiple identities to other nodes in the network. The additional identities are called as Sybil nodes. In this paper, few number of Sybil nodes are introduced into the network. The attack is modeled in such a way that the Sybil nodes communicate directly with the legitimate nodes. When the legitimate node sends some information to the Sybil Identity, it means that the malicious device or adversary overhears to the messages and keying information. Similarly the messages sent from the Sybil nodes are actually sent from the adversaries that presents multiple identities. In this way the adversary can also steal the identity of a legitimate node and after creating a false identity, it would start compromising the nodes. At this point, we are measuring the resilience of the network by generating keys using point addition and point doubling property of ECC, which prevents the adversary from compromising the nodes even after it would have got the identity of the legitimate nodes.

Algorithm:

```
x coordinate value;
y coordinate value;

Initialize:
communication range;
threshold value;
total number of malicious nodes;

for i=1:number of nodes
      distance= sqrt((node(x)-malicious node(x))^2 +
                (node(y)-malicious node(y))^2)
if (distance<communication range)
for i=1:number of nodes
for j=2:number of  nodes
     distance1= sqrt((nodei(x)-malicious nodej(x))^2 +
                (nodei(y)-malicious nodej(y))^2)
if (distance1< communication range)
neighborcount++
end
end
end
sybilnodecount++
if (flag= =0)
if(probability of malicious node compromising
  a legitimate node = = 0.5)
   node capture through Sybil function++
end
end
end
end
if(node capture through sybil function> threshold)
   ntw_capture through closest neighbor function++
end
```

4 Simulation Results and Discussions

In this section, the performance of the proposed scheme is evaluated by comparing it with the existing schemes. The performance of the network is analyzed in terms of connectivity and resilience.

Table 1 Seed key points $E_{107}(9,17)$

1,53	1,54	4,44	4,63	7,40	7,67	10,12
12,26	12,81	14,31	14,76	18,33	18,74	19,29
23,34	23,73	24,19	24,88	27,24	27,83	29,48
33,37	33,70	34,12	34,95	35,34	35,73	38,33
43,14	43,93	45,13	45,94	46,41	46,66	49,34
54,28	54,79	62,20	62,87	63,12	63,95	64,42
68,38	68,69	71,44	71,63	74,22	74,85	76,11
80,10	80,97	81,41	81,66	82,18	82,89	83,23
86,25	86,82	87,41	87,66	88,7	88,100	90,27
93,6	93,101	94,48	94,59	96,37	96,70	99,17
11,22	22,22	32,44	42,15	51,33	65,39	79,47
91,48	100,50	11,85	22,85	32,63	42,92	51,74
79,60	85,70	91,59	100,57	10,93	19,78	29,59
38,74	49,73	64,65	76,96	83,84	90,80	99,90
85,37	65,68					

4.1 Seed key generation

The seed points are generated according to the elliptic curve (1). In order to generate the seed keys, the value of prime integer should be chosen greater than the number of nodes in the network. The seed keys are generated according to the number of nodes in the network. Table 1 provides the generated seed key values for hundred numbers of nodes. The value of *p*, *a*, *b* are chosen based on the network requirements and satisfying the condition according to (2). For a network with hundred nodes, the value of prime *p* is chosen to be 107 which is the next highest prime number to 100 that could generate a minimum of 100 seed points which is not the case for prime 103. A unique seed key is assigned to each node in the network.

4.2 Private key ring generation

For each sensor node a key ring size of *R* is generated from the seed point. If two points are different then point addition operation is used. If the two points are similar point doubling arithmetic operation is used. The private key ring generated for seventeen numbers of nodes through simulation is shown in Table 2.

4.3 Secure Link Formation

A wireless sensor network consisting of 100 nodes has been simulated onto a deployment area of 50 x 50 square meters. The area of interest is divided into equal number of hexagonal grids. Generally the deployment is via air-borne method. Here the geographic location of each sensor node is known to the neighboring node as hexagonal deployment models are used. The deployment scenario is depicted in Fig.3

Each elliptic curve point is considered as the unique seed key for each and every node in the network. Using the elliptic curve operation such as point addition and point doubling over the seed keys the private key rings are generated for each node. Once the nodes are deployed into the field, each node in the network, try to find a common key to establish a secured link between them. A link is formed between a pair of nodes only if they share a common key from the private key ring. It is also required that the nodes are within the communication range. Once these conditions are satisfied the nodes in the network establish a secured communication link as shown in the Figure 3.

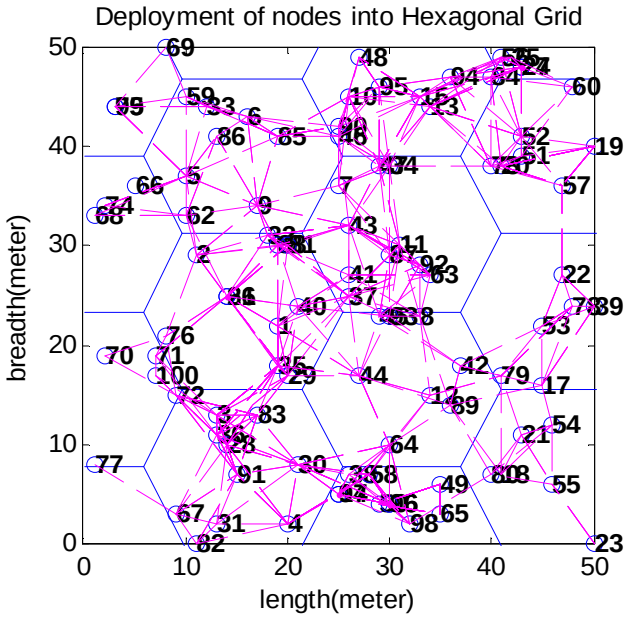

Fig.3. Secure link formation

4.4 Connectivity analysis

Local Connectivity is the probability of two nodes that shares at least one key to form a secure connection based on the key ring size. The connectivity analysis is performed for the Proposed Scheme, existing deployment knowledge based DDHV Scheme and a comparison is made.

4.4.1. Connectivity analysis of existing scheme

The deployment knowledge based scheme improves the local connectivity. It is clearly observed that the deployment knowledge based DDHV scheme significantly improve the local connectivity of their counterparts. As the nodes are divided into groups and each group is given a subset of key pool. Since nodes pick keys from a smaller pool, less memory is needed to achieve even a higher connectivity. The key pool size $|S|$ is fixed; the only parameter that can affect the local connectivity is τ [6], the number of key spaces carried by each sensor. The scheme improves connectivity and reduces unnecessary key assignments when compared to other existing schemes. At the same time the connectivity is achieved only 56% by using the lesser memory as shown in Fig.4

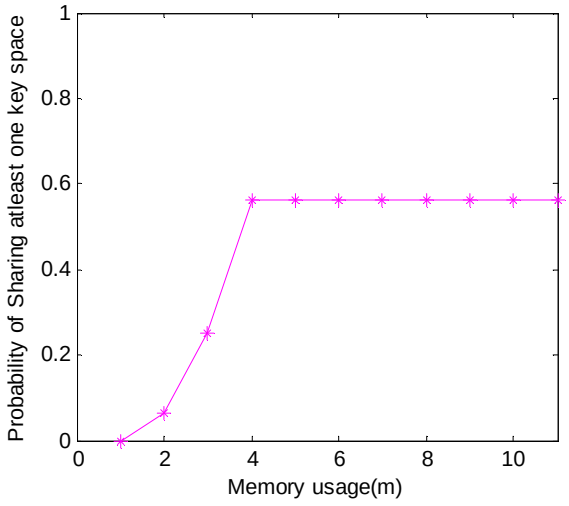

Fig .4. Connectivity analysis - DDHVD

Table 2 Key ring values

	R=1	R=2	R=3	R=4	R=5	R=6	R=7	R=8	R=9	R=10
Node 1	1,53	35,34	64,42	22,22	18,74	4,44	4,63	18,33	22,85	64,65
Node 2	1,54	35,73	64,65	22,85	18,33	4,63	4,44	18,74	22,22	64,42
Node 3	4,44	1,54	18,74	35,73	22,22	64,65	64,42	22,85	35,34	18,33
Node 4	4,63	1,53	18,33	35,34	22,85	64,42	64,65	22,22	35,73	18,74
Node 5	7,40	96,70	90,27	35,34	11,22	29,59	87,66	22,22	88,100	99,90
Node 6	7,67	96,37	90,80	35,73	11,85	29,48	87,41	22,85	88,7	99,17
Node 7	10,12	7,40	82,18	96,70	54,28	90,27	76,96	35,34	65,68	11,22
Node 8	10,95	7,67	82,89	96,37	54,79	90,80	76,11	35,73	65,39	11,85
Node 9	11,22	99,90	27,24	64,65	45,13	93,6	32,63	4,63	87,41	96,37
Node 10	11,85	99,17	27,83	64,42	45,94	93,101	32,44	4,44	87,66	96,70
Node 11	12,26	63,95	79,47	99,17	81,66	24,88	65,39	64,42	14,31	7,40
Node 12	12,81	63,12	79,60	99,90	81,41	24,19	65,68	64,65	14,76	7,67
Node 13	14,31	80,10	54,79	49,34	91,48	11,85	85,70	1,53	86,25	88,100
Node 14	14,76	80,97	54,28	49,73	91,59	11,22	85,37	1,54	86,82	88,7
Node 15	18,33	64,42	35,73	4,44	1,53	22,85	22,22	1,54	4,63	35,34
Node 16	18,74	64,65	35,34	4,63	1,54	22,22	22,85	1,53	4,44	35,73
Node 17	19,29	11,22	46,41	99,90	33,70	27,24	43,93	64,65	76,11	45,13

4.4.2 Connectivity analysis of proposed scheme

For each sensor node a key ring size of *R* is generated from the unique seed point. Through mathematical analysis, it is found that the private key ring consists of more than one common key to establish a secured link with other nodes. In other words *R* number of common keys is found. Thus each node is capable of establishing a secured link among *R* number of nodes. So each node in the sensor network has a complete connected graph. Therefore the connectivity achieved by the proposed scheme is 100% thus outperforming the existing deployment knowledge based schemes.

In the proposed work, the key ring is generated for 100 numbers of nodes for a key ring of size 10. Table.2 shows the key ring values for 17 numbers of nodes through simulation. It is observed that each and every node can find more than one common key between any pair of nodes, which can be easily understood by an illustration. Consider the first node which contains 10 secret keys in its memory. The similar coloured key values from Table.2 represent the common keys present among the corresponding pairing nodes to establish the connected graph. Thus the connectivity is improved to 100 percentage in the proposed scheme when compared to all other schemes in the literature.

4.5 Resilience analysis

Already discussed attack models are introduced in to the network. The malicious nodes try to capture the entire network by attempting to crack the private keys of the legitimate nodes through various network attack patterns. If they succeed in accessing the private keys of the nodes, they would be able to establish a link to communicate with the legitimate nodes and thereby gain access to the information carried by them.

Resilience is the ability of the network to maintain the security level even some nodes are captured. This is done for three important attacks namely random attack, closest neighbor attack and sybil attack. A comparison of the resiliency of the networks formed using the proposed scheme and other existing scheme against this attack is done. For resilience analysis, different number of malicious nodes can be set to run the simulation for a number of times and finally the number of times the network captured is recorded. It is clearly shown that there is a clear distinction between the performances of the two networks for the given attacks. The one formed by the proposed scheme is found to show better resiliency.

Random attack:

When the malicious nodes are introduced into the network by the attacker, each and every malicious node randomly pick a legitimate node based on the condition discussed in the algorithm to break the private keys of the sensor network. Once the private key is found, then that legitimate node is said to be compromised.

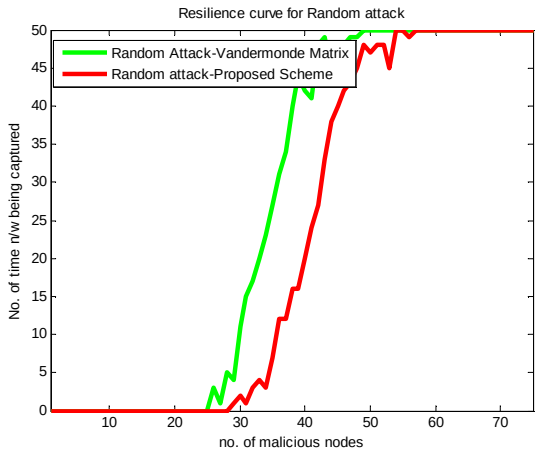

Fig.5. Random attack – resilience analysis

In the Fig.5 the number of times the network is being captured versus the number of malicious nodes introduced for both schemes are depicted. For around 25 numbers of malicious nodes, the network starts getting compromised and the introduction of around 49 malicious nodes causes the entire network to be captured in the case of the existing scheme. While around 58 malicious nodes are needed to capture the entire network for proposed scheme i.e. for the same number of malicious nodes introduced, a network employing key distribution using the Vandermonde matrix is less secure when compared with the network using proposed scheme.

Closest neighbor attack

In closest neighbor attack the malicious nodes are introduced by the attacker to compromise the network. The malicious node tries to compromise the legitimate node in the network to compromise the keys found in the memory of the sensor node as discussed in attack model. When the legitimate node is compromised, the compromised legitimate node tries to compromise the nearest neighboring node which is again a legitimate node.

It can be seen from Fig.6 that the introduction of around 30 malicious nodes causes the entire network to be captured for the existing key predistribution scheme. But for the proposed scheme around 38 malicious nodes are required to compromise the entire network. Thus the resilience

is improved in the proposed scheme when compared with the existing key predistribution scheme.

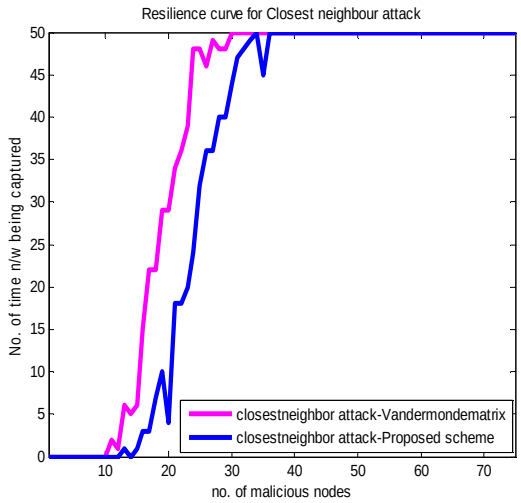

Fig.6. Closest neighbor attack-resilience analysis

Sybil attack:

Sybil attack is a harmful threat to the wireless sensor network as it affects the entire network by forging the identities from the legitimate node and so the malicious nodes malfunction as nodes with distinct identities. The attack is modelled based on previously discussed algorithm as in section (3.3)

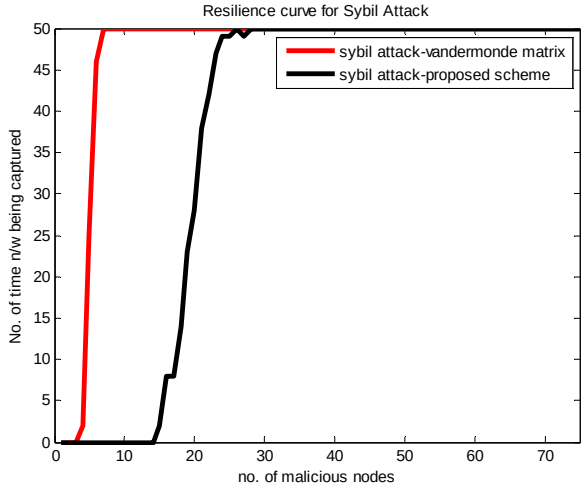

Fig.7. Sybil attack –resilience analysis

The Fig.7 depicts that the network starts getting compromised when the malicious node count is 3 and the entire network is captured for the malicious node count equal to 7 for the existing scheme. In the proposed scheme, the network starts getting captured when 14 numbers of malicious nodes are introduced; the entire network gets captured when the malicious node count reaches 27. The malicious node gets the access to the shared common key and uses the common key for further communication with the legitimate node as a trusted node. By iteratively repeating the process, the malicious nodes could capture the entire network by using the false identity. Thus Sybil attack poses a serious threat to wireless sensor network which affects the security of the network.

In all the above attack patterns discussed for the proposed scheme, the malicious nodes will try to find the corresponding seed key to compromise the corresponding node. But it is difficult to find the seed keys that belong to a unique elliptic curve group, which is based on the parameters *a*, *b* and *p*, as all these values depends on the network parameters and also it is a long time process for the attacker to capture a single node. It is also observed that a single node communicate with other node using more than one common key. So the proposed scheme outperforms the other schemes mentioned in the literature.

4.6 Memory requirement

Another major performance parameter in wireless sensor network is the memory requirement as it is a memory constrained environment. It depends on the number of keys stored in each node. As the size of the sensor node is very small, it consists of lesser memory space to store the key values. Therefore there exists a trade-off between the sensor node memory and sensor energy. With the increase in number of keys stored in each sensor node, the energy consumption also increases. Therefore the keys stored in each sensor node play a vital role in sensor memory, energy and connectivity.

The connectivity of the sensor network also depends on the key ring size. Therefore the key ring size should be chosen in such a way that the connectivity of the network is not affected. For a connectivity probability value of 0.33, EG scheme with deployment knowledge based key predistribution scheme requires 200 keys to be stored in each node. For DDHV scheme with deployment knowledge 46 keys need to be stored in the node achieve the same probability connectivity of 0.33 [6]. Similarly to achieve a connectivity probability value of 0.5, EG scheme with deployment knowledge requires 263 keys and DDHV-D scheme requires 67 keys as shown in Fig.8 and Fig.9.

In the proposed scheme whatever may be the number of nodes in the network the key ring size is fixed as 10 as it offers a 100% connectivity. Literature also proved that the ECC keys are smaller in size when compared to other algorithms. Eventhough the keys are smaller in size it provides the same level of security provided by other

algorithms. From the Fig.8 and Fig.9 it is observed that the proposed scheme requires lesser memory storage which is better for resource constrained devices like sensor node.

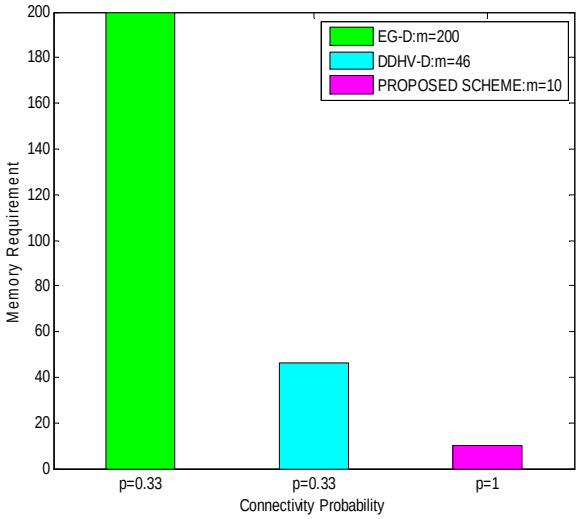

Fig.8. Memory requirement

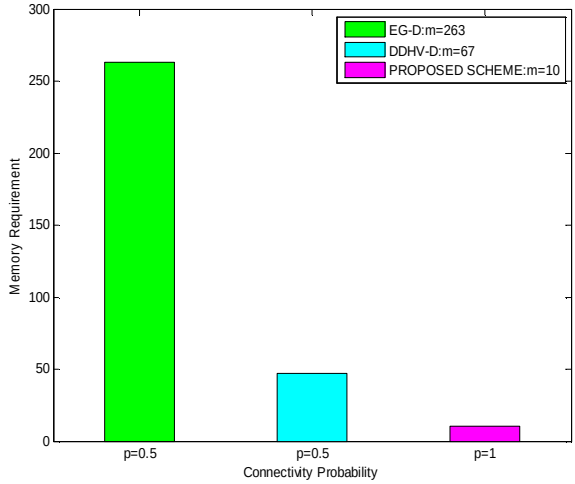

Fig.9. Memory requirement

5 Conclusions and Future work

The proposed scheme using ECC for key generation considers the resource constraints like memory, energy and provides better performance in terms of connectivity and resilience. The key ring size is small and fixed for each node which requires lesser memory, as the keys are generated using Elliptic Curve Arithmetic operations and by making use of deployment knowledge. Even though the key ring size is smaller than the existing schemes; the proposed scheme is able to achieve 100% connectivity. In terms of security, as each and every key stored in the node belongs to a unique elliptic curve group, it is difficult for the attacker to compromise the key. This improves the resilience of the network. Normally deployment knowledge based key predistribution scheme is better for resource constrained environments and it improves the resilience and connectivity in wireless sensor networks. Since Elliptic Curve Cryptography uses shorter key size which is suitable for power constrained environments the energy requirements are reduced in the wireless sensor networks. In the future work the resilience of the network can be further improved by encrypting the communication between the nodes.

References:

[1] Ian F.Akyildiz I, Su W, Sankarasubramaniam Y, and Cayirci. E), "A survey on sensor networks,"IEEE Commun. Mag., vol. 40,2002, pp. 102–114.

[2] Blom R ''An Optimal Class of Symmetric Key Generation Systems,'' in Advances in Cryptology EUROCRYPT '84, LNCS, vol.209, Springer-Verlag,1985, pp. 335–338.

[3] Eschenauer .L and Gligor V D," A Key-Management Scheme for Distributed Sensor Networks," In Proceedings of the 9thACMconference on Computer and Communications Security- CCS'02,2002, Washington D.C., USA, pp. 41-47.

[4] Du w, Deng J, Han Y S and Varshney P K , "A Pairwise Key Preistribution Scheme for Wireless Sensor Networks," in ACM CCS'03,2003.

[5] Du W, Deng J, Han Y S, and Varshney P K , "A Key management SScheme for Wireless Sensor Network Using Deployment knowledge," proc.IEEE INFOCOM'04,2004,pp.586-597

[6] Du W, Deng J, Han Y S and Varshney P K, "A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge,"IEEE transactions on Dependable and Secure Computing,Vol.3,2006,pp.62-77.

[7] Zhen Yu, Young Guan, "A Key Predistribution Scheme Using Deployment Knowledge for Wireless Sensor Networks,"Fourth International Symposium on Information Processing in Sensor Networks,2005,pp.261-268.

[8] William Stallings,Cryptography and Network Security –Principles and Practices, 3rd ed.Upper saddle River, NJ: Prentice Hall,2003.

[9] Koblitz N, "Elliptic Curve Cryptosystems," Mathematics of Computation, 1987, vol.48, pp.203-209.

[10] Miller V, "Uses of elliptic curves in cryptography," Advances in cryptology : proceedings of Cypto'85, 1986, pp.417-426.
[11] Wang Y, Attebury G, and Ramamurthy B, "A survey of security issues in wireless sensor networks," IEEE Commun. Surveys Tutorials, vol.8, 2006 pp. 2–23.
[12] Padmavathi G,Shanmugapriya D, "A Survey of attacks, Security Mechanisms and challenges in Wireless Sensor Networks," International Journal of Computer Science and Information Security, 2009,vol.4,pp.118-125.

Kishore Rajendiran graduated from Madras University, in Electronics and Communication Engineering during the year 1998. He obtained his Master degree in Communication Systems from Pondicherry Engineering College, Pondicherry University, Pondicherry during the year 2003 and PhD in the area of Security Issues in Wireless Sensor Networks from Anna University, Chennai. At present he is working as Associate Professor in the Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, Tamilnadu, India. His current area of research is security issues in Wireless Sensor Networks.

Radha S graduated from Madurai Kamraj University, in Electronics and Communication Engineering during the year 1989. She obtained her Master degree in Applied Electronics from Government College of Technology, Coimbatore and PhD degree in the area of Mobile Ad Hoc Networks from Anna University, Chennai. At present she is working as Professor & Head in the Department of ECE, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, India. She has 33 publications in International and National Journals and conferences. Her current areas of research are security and architecture issues of mobile ad hoc networks and sensor networks. She received IETE – S K Mitra Memorial Award in October 2006 for the Best Research Paper published in the IETE Journal of Research and **CTS – SSN Best Faculty Award – 2007 & 2009** for the outstanding performance for the academic year 2006-07 & 2008-09.

L.Cherlyflar graduated from Noorul Islam College of Engineering, in Electronics and Communication Engineering during the year2008. She obtained her Master degree in Communication Systems from Sri Sivasubramaniya Nadar College of Engineering, Chennai, during the year 2011. At present she is working as Software Engineer in HCL Technologies, Chennai, India.