Journal of Computer Science, 9 (5): 543-550, 2013
ISSN 1549-3636

doi:10.3844/jcssp.2013.543.550 Published Online 9 (5) 2013 (http://www.thescipub.com/jcs.toc)

OVERVIEW ON KEY DISTRIBUTION PRIMITIVES IN WIRELESS SENSOR NETWORK

[1]Raghini, M., [2]N. Uma Maheswari and [3]R. Venkatesh

[1]Department of Information Technology, K.L.N College of Engineering, India
[2]Department of Computer Science,
[3]Department of Information Technology,
P.S.N.A. College of Engineering and Technology, India

Received 2012-04-28, Revised 2012-06-28; Accepted 2013-05-17

ABSTRACT

Owing to the security requirements of wireless sensor network, the background of Wireless Sensor Network (WSN) is to be analyzed with different threats and attack models. Physical compromising of sensor nodes by an adversary is an emerging problem in sensor network and accordingly, it is necessary to provide an environment with efficient key management techniques due to resource constraints on sensor network. It is obvious to evaluate the efficiency of symmetric key management schemes for WSN, since it is not feasible to use traditional key management techniques such as asymmetric key cryptosystem and Key Distribution Center (KDC). This survey paper aims to report an extensive study on classification of pairwise key pre-distribution techniques. Further a smaller portion of analysis and security issues using pairwise key management is pronounced. Analysed results shows that polynomial pool based method have higher probability of communication by non-compromised nodes when compared with other schemes. The proposed survey effectively track the merits and demerits of different key predistribution schemes, also the communication overhead and memory overhead is reduced in polynomial pool based method during execution.

Keywords: Key Distribution Center (KDC), Hierarchical WSN (HWSN), Distributed WSN (DWSN), Base Station (BS), Symmetric Key Management Schemes

1. INTRODUCTION

Sensor network in today's world is used in variant fields such as medical appliances, military forces, wildlife tracking system, environmental monitoring and traffic control application, weather checking and regularity checking of temperature. The main purpose of WSN is to serve as an interface to the real world. WSN offers the major services such as monitoring, alerting and provisioning of information. The sensor node which is battery powered have the capability of collaborating with other nodes in terms of sharing and tranceiving information and also it can act independently in order to operate autonomously (Chakrabarti *et al*., 2006; Chandramathi *et al*., 2007). In sensor network security, an important challenge is the design of protocols to bootstrap the establishment of a secure communications infrastructure from a collection of sensor nodes which may have been pre-initialized with some secret information but have had no prior direct contact with each other. We refer to this problem as the bootstrapping problem. A bootstrapping protocol must not only enable a newly deployed sensor network to initiate a secure infrastructure, but it must also allow nodes deployed at a later time to join the network securely. The difficulty of the bootstrapping problem stems from the numerous limitations of sensor networks. Some of the more important ones include the inability to utilize existing public key cryptosystems (since the expensive computations involved could expose the power-constrained nodes to a denial-of-service attack), the inability to pre-determine which nodes will be neighbors after deployment and the inability of any node to put absolute trust in its neighbor (since the nodes are not tamper resistant and are vulnerable to physical capture). Also security is a critical issue when sensor networks are

Corresponding Author: Raghini, M., Department of Information Technology, K.L.N College of Engineering, India

deployed in a hostile environment where they are exposed to a variety of malicious attacks. For example, an adversary can easily capture sensors, impersonate a mobile sink, or provide misleading information.

The two basic architecture on wireless sensor network are Distributed WSN (DWSN) and Hierarchical WSN (HWSN). In distributed configuration, all the nodes will be participating in decision making process. In hierarchical configuration, the network model will be divided into clusters or groups of nodes where the cluster head will involve in decision making process like data aggregation. The data flow in DWSN is similar to HWSN with the only difference that the broadcasting can be done by every sensor node in DWSN **Fig. 1**. Here, the Communication pattern falls into three categories such as node-node communication (aggregation of sensor reading), node-Base Station (BS) communication (sensor readings), base station-node communication (specific requests) (Akyildiz *et al*., 2002; Chandramathi *et al*., 2007). The security requirements in any type of WSN will concentrate on confidentiality, authentication, integrity and others. However, achieving this goal is not an easy task in WSN and thus it is essential to sustain security in such type of network with an efficient key management scheme. This survey focuses on key management schemes on DWSN since there is no fixed infrastructure and the network topology is also not known until deployment. The only configuration is that, the sensor nodes are scattered randomly inside the target area and after deployment all sensor nodes scan its radio coverage area to notify its neighbors. Further section explains about threats in WSN, Security Primitives, Pairwise key predistribution and Polynomial pool based.

1.1. Attributes of Sensor Networks

1.1.1. Sensors

- Size: Small (MEMES), large (radar, satellites)
- Type: Passive (seismic, video), active (radar)
- Composition or mix: homogeneous, heterogeneous
- Spatial Coverage: dense, sparse
- Deployment: Fixed and planned (factory networks),ad-hoc(air dropped)
- Dynamics: Stationary (seismic), mobile(robot vehicles)

1.2. Sensing Entities of Interest

- Extent: Distributed (environmental), localized (target tracking)
- Mobility: Static, dynamic
- Nature: Co-operative (air traffic), non-cooperative (military targets)

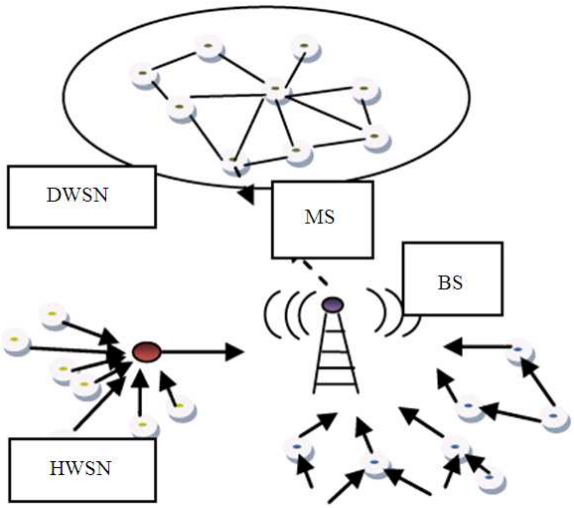

Fig. 1. Architecture of wireless sensor network

1.3. Communications

- Networking: Wired, wireless
- Bandwidth: High, low

1.4. Processing Architecture

- Centralized, distributed

1.5. Threats in WSN

WSN face multiple threats that affect the functionality and benefits of nodes. These threats can be categorized as denial of service attack, node compromise, protocol specific attack, common attacks, impersonation attack and others. The new challenge in WSN is node compromising since compromising any single node will affect the whole network. Various complex attacks can be easily emerged from compromised nodes in which the most familiar attack is the impersonation attack, where the compromised nodes make use of the non-compromised node's identity to perform active attack (Karlof and Wagner, 2003).

1.6. Security Primitives

The basic security primitives for the sensor node are to provide minimal protection to the data flow and need of secure protocol in order to avoid bootstrapping problem. The fundamental primitives are symmetric key cryptography, public key cryptography and hash primitives. Due to secure communication infrastructure from a collection of sensor nodes which may have been

pre-initialized with some secret information have no prior direct contact with each other and this leads to bootstrapping problem. Symmetric key cryptography primitives share the same secret key for origin and destination and it provides confidentiality, integrity and authentication. Symmetric algorithms are not very complex and they can be implemented easily in resource constraint devices. Hash primitives compress a set of data variable length into a set of bits of fixed length and it provides integrity of the information flow.

Hash functions are resource heavier and ten times slower than symmetric key functions. Public key cryptography primitives share two different keys for origin and destination. It has small size of keys hence it is memory and energy saving but the computational cost is high. Comparatively symmetric cryptography primitives are much desirable for WSN (Eschenauer and Gligor, 2002).

1.7. Issues IN Distributing Keys

In some of the applications, all the data transmitted through the network are critical and secure communication is needed for them. So cryptographic key management is a challenging task in wireless sensor networks. But sensor networks have some characteristics which make it difficult to communicate securely. Some of those characteristics are listed below:

Generally sensor networks consist of large number of sensor nodes which makes it difficult to secure each and every node. Sensor nodes are very inexpensive tiny devices and most of the time they are kept unattended. That makes them a victim of physical attack:

- Sensor nodes are constrained in resources which makes difficult to implement complex cryptographic algorithms. Because of constrained resources it is difficult to implement public key cryptography in sensor networks
- Wireless nature of the networks makes it easier to eavesdrop. There is no definite network topology in sensor network. Because of that it is difficult to implement any protocols

Because of these challenges, key establishment is a very challenging task in sensor network. Key establishment via a trusted center through secure channel is difficult to implement because of it is too costly. So, we generally use key pre-distribution as a procedure to establish keys in case of sensor networks. Key pre-distribution is a mechanism in which keys for each node are chosen from a large key pool.

1.8. Support For Security Primitives

1.8.1. Key Management Schemes

Key is used for secure communication among two or more sensor nodes, the group of keys stored in sink node is named as key pool and the collection of keys in every other sensor node is known as key ring. The common key types for WSN are node key- the key which is shared by a node and the base station, link key- it is a pairwise key which is shared by neighbors, cluster key-the key which is shared by a node and all its neighbors and network key-the key which is shared by all nodes and the base station.

To generate keys and distribution of those keys will be done by using key management system. There are three basic factors in key management system, such as key storage, key distribution and key maintenance. The two different cases for key management system are global keying and pairwise keying. Single key is generated for entire network and all the secure communications must be encrypted with the same key in global keying, whereas in the other case, every node should maintain the key for every other node within the network. Pairwise key improves network resilience against node capture and it enables authentication. Thus the second case is more secure because in global keying, any tamper node will release the global secret key and in return attackers will intrude the communication (Eschenauer and Gligor, 2002).

1.9. Pairwise Key Pre-Distribution

1.9.1. Random Key Pre-Distribution

Eschenauer and Gigor (2002) initially proposed a scheme called random key predistribution in which prior to the network deployment, a large pool of random keys are generated at the server and for each sensor node, the server randomly selects the subset of keys called key ring. When two sensor nodes want to communicate, at least one common key should exist in their key rings which are known as pairwise key. To reduce the fraction of compromised links between non-compromised nodes in random key predistribution, another scheme was developed which is Q-Composite Key Predistribution where instead of selecting a single common key, a Q number of common keys are selected.

1.10. Q-Composite Key Pre-Distribution

Instead of sharing a single key, the neighboring nodes share Q keys and use the hash of the Q keys as the shared key. Advantage: More secure against small-scale node capture as compared to the basic scheme.

Disadvantage: Not scalable. One approach to increase the resilience of the basic scheme against node capture attacks is to use q-composite random key pre-distribution as proposed in (Chan *et al*., 2003). The q-composite scheme differs from the basic scheme in requiring the nodes to have at least q common keys in their key rings in order to be able to establish a pairwise key. The pairwise key is then computed as the hash of all shared keys.

Essentially, the q-composite scheme degenerates into the basic scheme when q = 1. Intuitively, when q >1, the probability that two nodes can directly establish a shared key is smaller than the same probability in the basic scheme for the same values of the parameters k and m, because it is less probable to share at least q keys than to share at least one. Thus, in order to maintain the same expected degree of the nodes after the direct key establishment phase (and hence, to ensure secure connectivity), either the size m of the key rings should be increased, or the size k of the key pool should be decreased.

However, neither of the above two options are desirable: in the first case, the memory use of the sensors is increased, whereas in the second case, an increased fraction of the keys in the pool is compromised by capturing the same number of nodes. It is true, however, that the latter effect (increased fraction of compromised keys) is counterbalanced by the fact that now, in order for the adversary to compromise a link, it must compromise all the keys that have been hashed together to obtain the link key.

The simulation results in (Chan *et al*., 2003) show that the q-composite scheme offers greater resilience against node capture than the basic scheme does only when the number of captured nodes is small, whereas it tends to reveal larger fractions of link keys when large number of nodes have been captured by the adversary. In effect, by requiring q to be greater than 1, we make it harder for the adversary to obtain sufficient information to compromise links at the beginning when only a few nodes have been captured. But once a certain amount of information is collected by capturing more nodes, it becomes more and easier to compromise further links. In other words, the q-composite scheme increases the entry cost of a node capture attack. This makes sense, as it is reasonable to assume that it is more difficult to capture a large number of nodes than to capture only a few of them.

1.11. Probabilistic Generation Key Pre-Distribution

Probabilistic generation key predistribution which is based on the random key predistribution. Here instead of generating a large pool of random keys, a key pool $|S_{pg}|$ is represented by a small number of generation keys. Prior to the network deployment, the server generates a pool of randomly generated keys from which key rings are developed by applying hash algorithm for each generation key and further the node communication takes place as similar to that of random key generation process (Chan *et al*., 2003).

For large networks, a probabilistic method is more efficient than a deterministic method. This mechanism results from the concept all the nodes in the entire networks are connected with the 0.9997 probability-almost fully connected- if the probability each node can establish a pair-wise key with its neighbor nodes is 0.33. A key ring is stored in each node before deployment (a key ring k is randomly selected from key pool $|S_{pg}|$ which is randomly selected from huge key space). A common key in both key rings of a pair of nodes is used as their pair-wise key. It guarantees enough resilience even though not perfect resilience, because the probability of breaking communication link is $k/|S_{pg}|$. Moreover, it supports the large scale networks.

1.12. Polynomial Pool Based Pre_Distribution

Another type of key predistrbution scheme is polynomial pool based key predistribution, in which the setup server generates a pool of t-degree bivariate polynomials. Then each polynomial share is consider as key ring and given to deployed nodes. Using the polynomial share each node is able to calculate the key. The bivariate polynomials are identified by some unique identifier. The further communication is continued as similar to above schemes. By viewing the above schemes, it is clear that the security is highly pronounced in polynomial pool based method since key is calculated from the polynomial share. Thus the methodology of polynomial pool based scheme is focused in detail in the upcomming description (Rasheed and Mahapatra, 2011).

1.13. Polynomial Pool Based Key Pre-Distribution

Pairwise key establishment in this scheme has three phases: setup, direct key establishment, path key establishment.

Phase 1

The setup phase is performed to initialize the nodes by distributing polynomial shares to them.

In **Fig. 2** each node which receives a polynomial share will calculate the corresponding key:

- Setup server randomly generates a set F of t-degree polynomials over the finite field Fq
- For each sensor node i, the server picks a subset of polynomials
- The server assigns the polynomial shares of these polynomials to node i

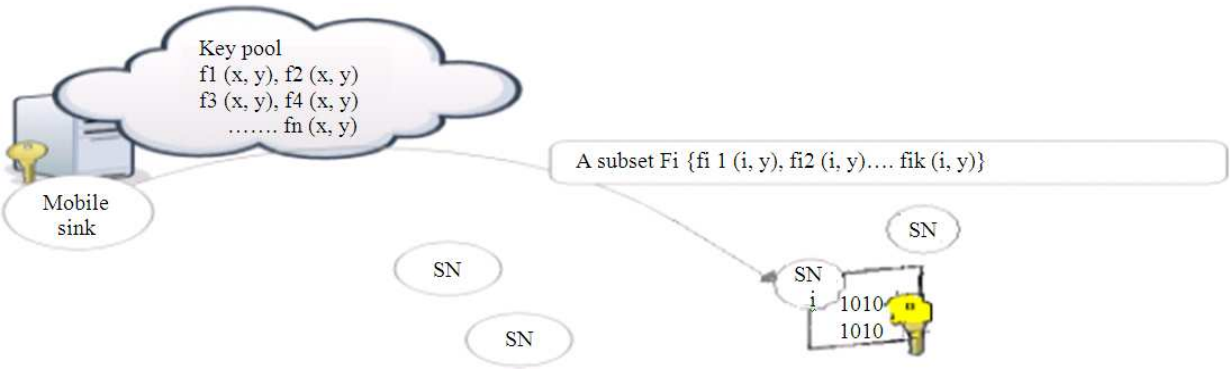

Fig. 2. Setup phase

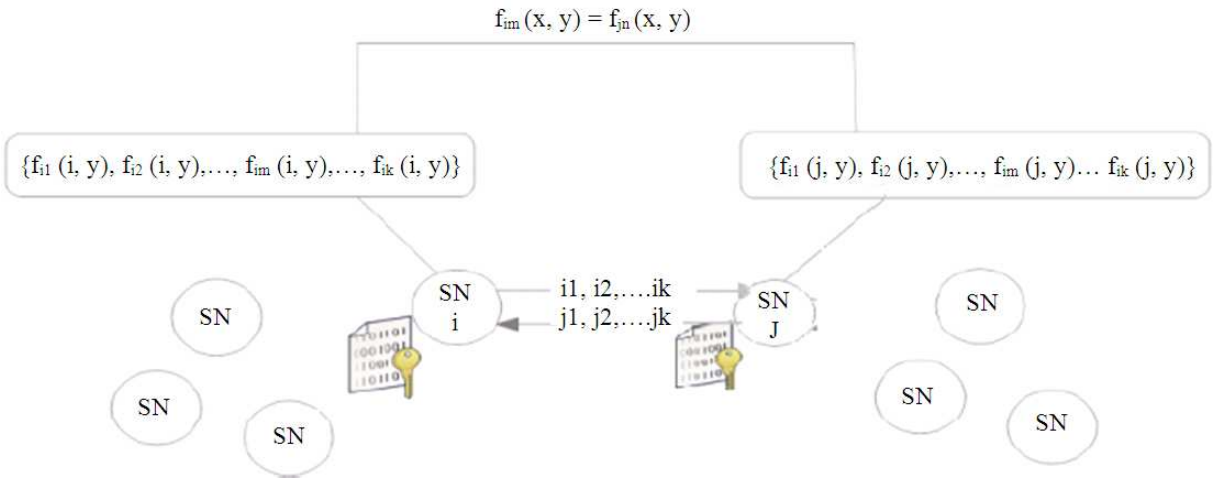

Fig. 3. Direct key establishment

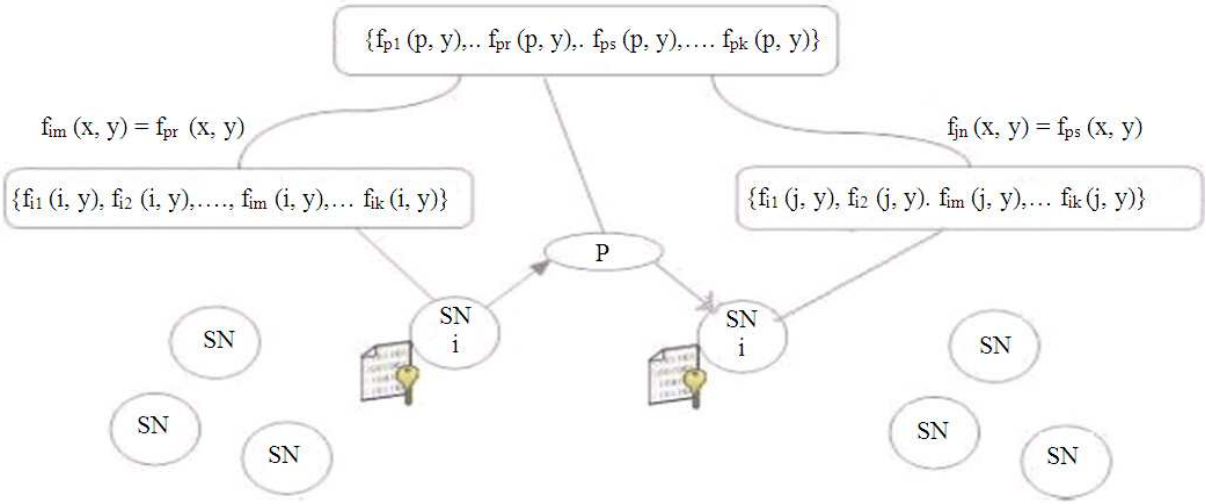

Fig. 4. Path key establishment

Phase 2

After network deployment, if any two sensor nodes need to establish a pairwise key, they first attempt to do so through direct key establishment. If both sensors have polynomial shares on the same polynomial, they can establish the pairwise key directly **Fig. 3**. Thus they can successfully establish a common key, there is no need to start path key establishment.

Phase 3

Node i and j cannot establish a key directly Node i needs to find a path between i and j. Any two adjacent nodes in the path can establish a pairwise key directly. Path discovery: Pre-distribution, Real-time discovery **Fig. 4**.

Step1:
Polynomial:
F (x, y) = [a+b (x+y)+cxy] mod P-(1)
For each node polynomial share:
g_u (x)= (an+bnx) mod P-(2)

Where:

An = (a+bru) mod P and bn= (a+cru) mod P-(3)

Step2:
For node u and v following computation are performed for communicate:

- Ku,v =Kv,u = f(ru,rv) =[a+b(ru+rv)+c rurv] mod P-(4)
  - U computes Ku,v= gu(rv)-(5)
  - V computes Kv,u= gv(ru)-(6)

Step 3:
$K_{v,u} = K_{u,v} = f(r_u,r_v) = [a+b(r_u+r_v)+c\ r_u\ r_v] \bmod P$-(7)
$K_{u,w} = K_{w,u} = f(r_u,r_w) = [a+b\ (r_u+r_w)+c\ r_u\ r_w\ ] \bmod P$-(8)
$K_{v,w} = K_{w,v} = f(r_v,r_w) = [a+b\ (r_v+r_w)+c\ r_v\ r_w\ ] \bmod P$-(9)

Result Shows that:

$K_{u,v} = K_{v,u}$
$K_{u,w} = K_{w,u}$
$K_{v,w} = K_{w,v}$

1.14. Security Analysis for Successful Communication of Any Two Nodes

3 nodes U, V, W

- Id = 12,7,1
- p = 17 (chosen parameter)
- a = 8,b = 7, c = 2 (chosen parameter)
- F(x, y) = [a+b (x+y)+cxy] mod P
- F(x, y) = [8+7(x+y)+2xy] mod P

1.15. G-Polynomial: Key Discovery

For U = 12:

$$g_u(x) = (an+bnx) \bmod P$$

For V = 7:

$$g_v(x) = (a=+b_nx) \bmod P$$

For W=1:

$$g_w(x) = (a_n+b_{nx}) \bmod P$$

Key generation:
$K_{v,u} = K_{u,v} = f(r_u,r_v) = [a+b\ (r_u+r_v)+c\ r_u\ r_v] \bmod P$
F (12, 7) = [8+7(12+7) +2(12)(7)] mod 17
= [8+133+168] mod 17 =309mod 17
$K_{v,u} = K_{u,v} = 3$

$K_{u,w} = K_{w,u} = f(r_u,r_w) = [a+b(r_u+r_w)+c\ r_u\ r_w] \bmod P$
F (12, 7) = [8+7(12+1)+2(12)(1)] mod 17
= [8+91+ 24] mod 17
=123mod 17
$K_{w,u} = K_{u,w} = 4$

$K_{v,w} = K_{w,v} = f(r_v,r_w) = [a+b\ (r_v+r_w)+c\ r_v\ r_w] \bmod P$
f (12,7) = [8+7(7+1)+2(7)(1)] mod 17
= [8+56+ 14] mod 17
=78mod 17
$K_{w,v} = K_{v,w} = 10$.

During the initialization phase each node has an unique id r_u which is unique and is a member of finite field Z_p . Three elements a, b, c are choosen from Z_p. Thus the polynomials f(x,y)=(a+b(x,y)+cxy) mod p is calculated. Thus **Table 1-3** shows how each node calculates its polynomial share with $g_u(x) = (a_n+b_{nx}) \bmod p$, where $a_n = (a+br_u) \bmod p$ and $b_n = (b+cr_u) \bmod p$.

In the above **Table 4-6**, nodes U, V and W computes their key using computed polynomial shares. The computed key value of nodes U and V matches each other and thus they can communicate efficiently. The same procedure is used for node pairs such as U-W and V-W.

1.16. Comparision of Conventional Methods

The various conventional methods available in pairwise key distribution is discussed in the below **Table 7**, where the pre-distribution methodology which uses pairwise key is dominated in the above survey which ensures performance in terms of their communication overhead and memory overhead.

Table 1. Polynomial share for node U

Where $a_n = (a+br_u) \bmod P$	Where $b_n = (b+cr_u) \bmod P$
=[8+7(12)] mod 17	=[7+2(12)] mod 17
=[8+84] mod 17	=[7+24] mod 17
=92 mod 17	=31 mod 17
an=7.	bn=14
gu(x)= (7+14x)mod 17	

Table 2. Polynomial share for node V

Where $a_n = (a+br_v) \bmod P$	Where $b_n = (b+cr_v) \bmod P$
=[8+7(7)] mod 17	= [7+2(7)] mod 17
=[8+49] mod 17	= [7+14] mod 17
=57 mod 17	= 21 mod 17
a_n = 6.	b_n=4
g_v (x)= (6+4x)mod 17	

Table 3. Polynomial share for node W

Where $a_n = (a+br_w) \bmod P$	Where $b_n = (b+cr_w) \bmod P$
= [8+7(1)] mod 17	= [7+2(1)] mod 17
= [8+7] mod 17	= [7+2] mod 17
= 15 mod 17	=9 mod 17
a_n = 15.	b_n = 9
g_w(x)= (15+9x) mod 17	

Table 4. Key generation For U-V

U computes	V computes
$K_{u,v} = g_u (rv)$	$K_{v,u} = g_u (r_u)$
= (7+14x) mod 17	= (6+4x) mod 17
= g_u (7) = (7+14 (7)) mod 17	= g_v () = (6+4(12)) mod 17
= [105] mod 17,	= [54] mod 17
$K_{u,v}$ = 3	$K_{v,u}$ =3
$K_{v,u} = K_{v,u}$ = 3	

Table 5. Key Generation For U-W

U computes	W computes
$K_{u,w} = g_u (rw)$	$K_{w,u} = g_u (ru)$
= (7+14x) mod 17	= (15+9x) mod 17
= g_u (1) = (7+14(1)) mod 17	= g_v () = (15+9(12)) mod 17
= [21] mod 17	= [123] mod 17
$K_{u,w}$ = 4	$K_{w,u}$ = 4
$K_{w,u} = K_{u,w}$ = 4	

Table 6. Key Generation For V-W

V computes	W computes
$K_{v,w} = g_v (rw)$	$K_{w,v} = g_u(rv)$
= (6+4x) mod 17	= (15+9x) mod 17
= g_v (1) = (6+4(1)) mod 17	= G_w (7) = (15+9(7)) mod 17
= [10] mod 17	= [78] mod 17
$K_{v,w}$ = 10	$K_{w,u}$ = 10
$K_{w,v} = K_{v,w}$ =10	

Table 7. Conventional methods

Pairwise Distribution	Mechanism	Keying style
Probabilistic	Pre-distribution	Random key chain Pairwise Key
Deterministic	Pre-distribution	Pairwise Key Combinatorial
	Dynamic key generation	Master Key Key Matrix Polynomial
Hybrid	Pre-distribution dynamic key generation	Combinatorial Key Matrix Polynomial

2. MATERIALS AND METHODS

The Polynomial Pool Based method is considered as an efficient one and the analysis of key generation process is done by calculating polynomial shares for each node and correspondingly the common key is found out between any two communicating nodes. The analysis was based on existing formula available in polynomial method but with different values for parameters to check whether the common key exist between any two nodes to communicate. Initially r_u, r_v, r_w is the unique id given to every participating nodes. For instance, here we have considered three nodes. Then polynomials are calculated using $f(x,y)=(a+b(x,y)+cxy) \bmod p$ with polynomial share $g_u(x)=(a_n+b_{nx}) \bmod p$, where $a_n = (a+br_u) \bmod p$ and $b_n = (b+cr_u) \bmod p$.

3. RESULTS

In this survey paper, we have given a proof of calculating common keys among two nodes which is efficient when compared to other key pre-distribution schemes. Since polynomial shares are calculated before generating keys, an adversary node which wants to match its key with non-compromised node's key is not an easy job. Thus security aspect and performance of polynomial method increases.

4. DISCUSSION

4.1. Discussion on Future Research Issues

The discussion on this survey is how to optimize further the key pre-distribution scheme. Several research directions are worth investigating. Even though polynomial pool based method enhances communication and memory overhead, But still it can only tolerate no more than t compromised nodes where t is limited by the memory available in the sensor nodes (Blundo *et al*.,

1993). Thus further optimization can be done by avoiding common keys between sensor nodes without compromising security. Research issues can focus on optimization on polynomial pool based method. Further study in the future is planning to evaluate security strength, according to kinds of active attacks. Since the communication overhead and memory overhead is already measured using the existing scheme, the future research work can also be in form of comparing active attacks vs communication and memory overhead, as a result the performance rate reaches several heights.

5. CONCLUSION

In this study, we have taken a survey on various key predistribution schemes for Distributed Wireless Sensor Network (DWSN). We have shown that the polynomial pool based scheme remains secure when compared to other schemes. Security analysis shown for polynomial pool based method using polynomial shares provide a higher probability for non-compromised sensors to establish a secure communication with mobile sink than previous schemes (Liu *et al*., 2003). Also communication overhead, memory overhead will be reduced during execution.

6. REFERENCES

Akyildiz, F., W. Su, Y. Sankarasubramanian and E. Cayirci, 2002. A survey on sensor networks. IEEE Comm. Mag., 40: 102-114. DOI: 10.1109/MCOM.2002.1024422

Blundo, C., A.D. Santis, A. Herzberg, S. Kutten and U. Vaccaro, 1993. Perfectly secure key distribution for dynamic conferences. Inform. Comput., 740: 471-486. DOI: 10.1006/inco.1998.2717

Chakrabarti, A., A. Sabharwal and B. Aazhang, 2006. Communication power optimization in a sensor network with a path-constrained mobile observer. ACM Trans. Sen. Netw., 2: 297-324. DOI: 10.1145/1167935.1167936

Chan, H., A. Perrig and D. Song, 2003. Random key predistribution schemes for sensor networks. Proceedings of the IEEE Symposium on Security and Privacy, (SP ‘03), ACM Press, Washington, USA., pp: 197-197.

Chandramathi, S., U. Anand, T. Ganesh, S. Sriraman and D. Velmurigan, 2007. Energy aware optimal routing for wireless sensor network. J. Comput. Sci., 3: 836-840. DOI: 10.3844/jcssp.2007.836.840

Eschenauer, L. and V.D. Gligor, 2002. A key-management scheme for distributed sensor networks. Proceedings of the ACM Conference Computer Communications Security, Nov. 18 - 22, ACM Press, New York, USA., pp: 41-47. DOI: 10.1145/586110.586117

Karlof, C. and D. Wagner, 2003. Secure routing in wireless sensor networks: Attacks and countermeasures. AdHoc Netw., 1: 293-315. DOI: 10.1016/S1570-8705(03)00008-8

Liu, D., P. Ning and R. Li, 2003. Establishing pairwise keys in distributed sensor networks. Proceedings of the 10th ACM Conference on Computer and Communications Security, Oct. 27-30, ACM Press, New York, USA., pp: 52-61. DOI: 10.1145/948109.948119

Rasheed, A. and R.N. Mahapatra, 2011. Key predistribution schemes for establishing pairwise keys with a mobile sink in sensor networks. Proceedings of the IEEE Transaction on Parallel and Distributed Systems, (TPTS ‘11), ACM Press, USA., pp: 174-184. DOI: 10.1109/TPDS.2010.57