Research Journal of Applied Sciences, Engineering and Technology 4(18): 3188-3200, 2012
ISSN: 2040-7467

Submitted: December 22, 2011 Accepted: February 17, 2012 Published: September 15, 2012

WSN: Key Issues in Key Management Schemes-A Review

D. Manivannan and P. Neelamegam
School of Computing, SASTRA University, Thanjavur-India

Abstract: The real challenges in the field of Wireless Sensor Networks (WSN) are to control the trade-off between providing security and conserving scarce resources. To provide security besides cryptographic primitives, secret key also play a vital role. The reason for key management is to set up and to provide secure channels among sensor nodes in WSN applications. Various key management schemes are proposed for WSN. Deploying sophisticated network based Key Management protocol in WSN is critical and important task. From the literature survey, it is understand that more energy is consumed to establish wireless sensor network in any real time applications. In this paper, various key issues in key management schemes are discussed based on the evaluation metrics such as energy responsiveness, key connectivity, resilience, scalability and efficiency.

Keywords: Distributed wireless sensor networks, group keying, hierarchical wireless sensor networks, lu matrix, network keying, pair-wise

INTRODUCTION

A wireless sensor network comprises large number of distributed, self-directed devices called sensor nodes also called as motes. WSN naturally encompasses a large number of spatially dispersed, petite, battery-operated, embedded devices that are networked to supportively collect, process and convey data to the users and it has restricted computing and processing capabilities. Sensor node consists of three major units like sensing, processing and transceiver Unit. The sensor field of WSN comprises of different n number of heterogeneous and homogenous nodes (Du *et al*., 2007) which are deployed in open surroundings and the interconnection among the nodes is established through wireless transceivers and that to be grouped with corresponding cluster heads and connected to gateway and to end with the base station. In homogenous Sensor Networks, all sensor nodes have the same property in terms of computation, communication, memory, energy level and reliability. In heterogeneous Sensor Networks, the mixture of different sensors has the different capabilities in terms of computation, communication, memory, energy level and reliability.

It's very important to evaluate the list of factors which is used to determine the performance of a sensor network. The general issues of WSN are data redundancy, addressing scheme, message size and resources. WSN's general security goals are; Confidentiality, Integrity, Authentication:, Availability, Survivability, Efficiency, Freshness and scalability. Wireless sensor networks are susceptible to many attacks because of its transmition nature, resource restriction on sensor nodes and deployment in uncontrolled environments. To ensure the security services in WSN many crypto mechanism are proposed. But the security strength of the entire cryptosystem of WSN mainly depends on the secret keys used, not in the algorithm. The reason of key management for WSN is to load, distribute and handle the secret keys in sensor nodes to establish a secure communications among sensor nodes. Security critical applications are depends on key management scheme because it has to provide high fault tolerance when a node get compromised. While designing the key management schemes, the important metrics to be evaluated: Local / Global Connectivity: each node communicates with every other node in the sensor field region. Resilience: whenever a sensor node is compromised, the key management scheme assured how secure the remaining communication link there will be i.e., resistances against node capture (Marcos *et al*., 2010) Scalability: capability to support when large numbers of nodes are added to the sensor network. Efficiency: In terms of storage, communication and computation.

In general key distribution scheme are widely classified into three categories. These are network, pair-wise and group keying. In network keying a single shared secret key is used for communication. This keying mechanism is excellent in terms of scalability and flexibility. But when one node gets compromised, entire network will gets compromised. In pair-wise scheme for n node, n-1 keys are stored, with the total n (n-1)/2 distinguishable keys in the network. Though it achieves perfect resilience, scalability is an issue. Group keying mechanism combines network keying and pair-wise keying mechanisms (Yang *et al*., 2007). Many Key management scheme are proposed for WSN are based any

Corresponding Author: D. Manivannan, School of Computing, SASTRA University, Thanjavur-India

one of the above keying mechanism. In this paperthe review is done based on key management evaluation metrics and finally summarized which key management scheme comes under distributed wireless sensor networks and hierarchical wireless sensor networks.

The key management schemes in wireless sensor networks are classified as follows; these are : Network-Wide Key distribution-BROSK, SKKE; Pairwise Scheme-full pairwise scheme, Probabilistic Based approach: EG scheme, Q composite scheme, Multipath Key Reinforcement Scheme, Random Pair-wise Key Scheme, key redistribution scheme and AP Scheme; Matrix Based: Blom's scheme, multi space key distribution scheme, LU Matrix and hierarchical LU matrix; Polynomial KMS: Blundo, polynomial pool based key pre-distribution scheme; Deployment Knowledge: Extending Probabilistic scheme, extending matrix based scheme; Group Keying: Tree based approach, EBS: SHELL, LOCK and LEAP.

In this study Network Wide Key Distribution Schemes, Pairwise key management schemes, Matrix based schemes, Polynomial based key management schemes, Key management with deployment Knowledge, Group keying schemes are discussed, finally summary of various key management schemes are done.

Network wide key distribution schemes: The general key distribution schemes in WSN are broadly classified into three types. They are network keying, pair wise keying and group keying. These classifications mainly concentrate on the number of keys being used in the system. The network keying scheme is very simple, manageable and involves the usage of very less resources. This method satisfies all the functional requirements like accessibility and self organization. A single key is used for all the nodes in the network. All the secured data processing in sensor nodes depends on a single secret key. All the nodes in the entire network use only a shared secret key. For example, in broadcast session key negotiation protocol (BROSK), proposed by (Lai *et al*., 2002), the secret session key for pair of nodes A and B is generated with the help of master key K, random nonce RN_A and RN_B and round function F. The secret session key $K_{AB} = F (K||RN_A||RN_B)$, Where F is pseudo random function (Marcos *et al*., 2010). Like that, in Symmetric-Key Key Establishment protocol (SKKE), key establishment involves in between initiator node and responder node. The common shared secret key K_{AB} is found with the help of master key K, random nonce RN_A and RN_B, Identifier, responder and round function F. The common shared secret session key $K_{AB} = F (K||RN_A||RN_B||ID_A||B)$, where F is pseudo random function. It is very flexible, scalable and self organized key distribution scheme. The main drawback of this scheme is the lack of robustness. Compromisation of any one node in the network will affect the security of the entire WSN.

PAIRWISE SCHEMES

Full pair wise scheme: Full pairwise scheme is a very basic and easily acceptable system. Each pair of node in the sensor field will have different key for data communication. Communication between any two nodes will take place only after the establishment of a common key between them. In this scheme, N-1 keys are required in each node where N is the total number of sensor nodes in the network. The total number of distinguishable keys in the network is N (N-1)/2. Using this scheme, each node can be authenticated and it also provides good robustness (Marcos *et al*., 2010). The main drawback of this scheme is its non-flexibility, non scalability and complexity.

Probabilistic approach:

EG scheme: This scheme proposes a simple key distribution in distributed wireless sensor networks. This scheme (Eschenauer and Gligor, 2002) consists of three basic steps they are:

- Key Pre-distribution phase
- Shared secret key discovery
- Path key establishment phase

Prior to Distributed WSN (DWSN) deployment, the first phase is ensured. In key initialization phase the large pool of P Keys is generated with its identifier. Then k Keys are selected from large pool P and stored in each sensor node memory unit. These k Keys along with it identifier is called key ring. After establishment of these steps, it is important to ensure that only less number of keys is placed on each sensor node and also ensure that any two nodes in the field share a common secret key. During second phase, the nodes have to find shared common key between its neighbors. Using simple broadcasting techniques with key identifier, the shared keys are discovered. It leads to general attacks by an adversary. This methodology provides easy loopholes to any misfeasor who wants to analyze or affect the sensor network. Alternate scheme for shared key discovery phase is generate random value α and encrypt with the keys in the key ring and then broadcast. The node which shares the common key will decrypt and get the value of α. So the node will communicate with that key.

In the path key establishment phase, at the end of shared key discovery phase some nodes do not have the shared key between the neighboring nodes, so communications between nodes are not possible. Such node will establish a path key. For new path key instead of generating a new key, the key ring will have some

unused keys. That key can be used as path key. These path key will be shared between nodes using the communication link which is discovered during shared key discovery phase.

Revocation: In DWSN, the entire sensor nodes in the sensor field must be controlled by controller node. There may be many controller node and group of sensor nodes comes under the control of controller node. Each node shares a secret key with the controller node. Controller node also maintains the information like the list of key identifiers each node is having. If any sensor node in the sensor field is compromised, revoking the entire key ring of that particular node is very important and also essential. Controller node broadcasts a single revocation message, which contains a signed list of key ring of compromised node. Controller node generates the signature key K_s for sign the list of key identifiers, then it encrypt with secret key which is shared with the nodes and unicast it to each and every node in the network. All the sensor nodes decrypt the signed message and it verifies the signature of the signed list and locates the corresponding keys and finally it will be removed. After revocation of the compromised key ring, reconfiguration of links is done by starting the shared key discovery phase followed by path key establishment phase. If the life time of the keys in the key ring expires, the self revocation of shared secret key by a node in the network is done.

Key connectivity is proportional to the sensor node memory. If the number of keys loaded into sensor node is high. Better key connectivity is obtained in this scheme. The advantage of this scheme is simple, efficient and scalable, than full pair-wise scheme. Communication cost also high because the node get compromised the revocation message is unicasted to every node in the network. This scheme fails to provide node to node authentication so this scheme is not suitable for high end security applications.

Q composite scheme: This scheme (Chan *et al*., 2003) is similar to basic schemes; the main difference is that EG scheme (Eschenauer and Gligor, 2002) requires single key to communicate between neighboring nodes but Q composite scheme requires q keys i.e. $q \geq 2$. The rest of the phases are similar to basic scheme i.e., generating the large key pool P and from the key pool randomly select m keys without replacement and load in to the sensor node. In Key sharing phase, each node found out the common keys with each of its neighbors. This can be done by each node by broadcasting its key identifier. Two nodes can communicate each other if both node have q' keys in common where ($q' \geq q$). Once common keys are discovered, shared secret key K between two nodes has to be established by taking hash of all common keys. i.e., for example K = Hash (k1||k2||…||kq'). They can be hashed in any order preferably in the order of key pool.

The major work involved in Q composite schemes is computation of key pool size. If the key pool size is large, the probability of sharing common keys between two nodes will be less. If the key pool size is reduced, even when small number of nodes is compromised, the attacker can get large keys of P. Based on this; the key pool size has to be chosen in such a way, the probability of sharing common keys should be in expected level and at the same time security level should also be maintained. With respect to security analysis, this scheme achieves perfect resilience against node capture compared to EG scheme (Eschenauer and Gligor, 2002). For example when q = 2, P = 0.33, m = 200: for this criteria if 50 nodes get compromised, the additional communication link compromised will be 4.74 in Q composite, where as 9.52 in EG scheme. But the drawback is if the large number of nodes compromised, there is a possibility of compromising the entire communication link. The advantage of this scheme is achieving better resilience but scalability is an issue. This scheme also fails to provide node-to-node authentication.

Multipath key reinforcement scheme: Chan et al proposed a scheme (Chan *et al*., 2003) which offers good security compared to EG scheme. In Q composite scheme the links are stronger than basic scheme because each link shares q keys (q>2). General problem in basic scheme is, when a link get compromised, there is a probability of breaking many communication links since keys are randomly selected from key pools, so probability of repeating same key will be more. When one node gets compromised, establishing new key using previously established link is unsecure. So link must be secure before updating the communication key. For that find out the multiple independent path from source to destination node. For example if node A wants to establish a new communication key with node B, first multiple disjoint path have to be find out say h paths. Then node A generates h random values and sends out in all possible disjoint path. Once after receiving, node B computes the new key. Simultaneously node A also computes the key as:

$$K' = K \oplus g_1 \oplus g_2 \ldots \oplus g_n$$

where, K is the original key from the key pool. Larger the value of h greater will be the security but leads to communication overhead and also finding the independent path from source to destination is also difficult. A Two hop approach is recommendable i.e., one intermediate node between source and destination. The main advantages of this scheme is that it achieves better security than basic scheme and Q composite scheme but there will be a communication overhead.

Random pair-wise scheme: This scheme (Chan *et al*., 2003) combines the features of full pair-wise scheme and EG scheme. Full pair-wise scheme achieves node-to-node

authentication, perfect resilience and perfect key connectivity. But storage is problem i.e., if n nodes are in the network then n-1 keys are loaded into sensor nodes and scalability is difficult. But in EG scheme storage can be reduced but fails to provide node to node authentication and perfect resilience. Chan *et al*. (2003) proposed a random pair-wise scheme in that for n nodes, m ($m \geq n$) unique identifiers are created. Each node has unique identity and is paired with K randomly selected identifiers. Pair-wise key is generated for each pair and is loaded into sensor node. Finally each sensor node is loaded with its identity,K identifier which is paired with its id and K corresponding pair-wise keys which is less than n-1 keys in fully pair-wise scheme. Once after loading shared key discovery phase starts, i.e., each node broadcasts its id. Nodes will find out the neighboring node which shares the same key. While identifying the neighbors, sometimes it may be beyond the communication range also, for that the node id will be rebroadcasted. But this rebroadcast should be limited to number of hops to avoid denial of service attack.

A random pair wise scheme adopts different mechanism for revoking the compromised node than basic scheme. In basic scheme, revocation is initiated by controller node. But here, any nodes which identify the compromised node will cast a vote against it and broadcast to all other node. If threshold t, vote has been casted, that node will get revoked. This is done to reduce the overhead of the base station. For voting each node is loaded with a key K_i and K-1 voting member of the hash values. So if K pair-wise keys are loaded in the node the storage requirement is O (K^2). Though random pair wise scheme achieves perfect resilience against node capture, scalability is an issue, because it can be extend till m-n nodes only. Storage also an issue because it has to store its identity and K identifiers, K pair wise keys, voting key K_i and K-1 hash values and threshold t value.

Key redistribution scheme: Law et al proposed a scheme (Law *et al*., 2007) with small changes in basic scheme i.e., it replaces the original path key establishment instead it includes key redistribution scheme. For example node A and node C share a common key K_1 and Node B and Node C share a common key K_2. There is no common key between Node A and Node B, so Node A does not communicate with Node B. In general Node A sends a request message to node C to get K_2. Node C encrypt K_2 value using K_1 and send to Node A. Node A will decrypt the received value using K_1 and get the K_2 and establish the communication between Node B. If suppose Node C refuses Node A request, the alternate approach is Node A select an unused key K_3 from the key ring and send to Node C. Node C computes new key K_{23}, where K_{23} = Hash ($K_2||K_3$) then K_{23} encrypted with K_1 and K_2 and send back to Node A and Node B respectively. Node A and Node B will decrypt the value using K_1 and K_2 and get the value of K_{23}. Now Node A and Node B will have the common key K_{23}, using that they establish the communication link between them.Compared with basic scheme, this Key Redistribution scheme provides higher key connectivity and also provides better resilience but the drawback is communication overhead.

AP scheme: In Basic scheme, to achieve higher key connectivity the number of keys loaded into sensor nodes should be high. For example if P is 10,000, each node should load more than 150 keys to achieve 0.9 key sharing probabilities. Since sensor nodes are resource constrained, loading large number of keys is difficult. To overcome this (Du *et al*., 2005) proposed a scheme for heterogeneous sensor networks called AP (asymmetric pre-distribution) scheme. Since heterogeneous it includes two different types of sensors called H sensors and L sensors. In terms of storage, computation and communication range H sensor will have strong capabilities than L sensors. H sensor and L sensors are scattered in the network. L sensor which comes within the communication range of H sensor will form a cluster. H sensor will act as a cluster head. As like basic scheme, AP scheme also undergo three phases. The phases are Key pre-distribution, shared key discovery and path key establishment. The difference is if any L sensor does not share any key with its neighbors H sensors obtains the key of both nodes compute the key and send to each node. The performance analysis of this scheme shows that it achieves better storage than EG scheme (Eschenauer and Gligor, 2002). For example if 1000 L sensor and 10 H sensors are there in AP, the keys loaded in L sensors is 20 and H sensor is 500 but for EG scheme, each L sensor is loaded with 150 keys to achieve the same key probability. This scheme works better in resilience because H sensors are tamper resistant and also compromising L sensor will not reveal much of the communication link because number of keys loaded in L sensor is less.

MATRIX BASED SCHEME

Blom scheme: In a perfect pair wire scheme, if the network consists of n users each user will have to store n-1 keys. If the size of the network increase number of keys stored in each nodes will be increased. Since sensor nodes are resource constrained implementing perfect pair wire scheme is difficult.

Blom proposed (Blom, 1985) a symmetric key generation system in which with the least information every node will communicate with every other node. Blom scheme is λ secure property i.e., the communication links will be secure until λ nodes get compromised .When λ nodes get capture the entire communication links in a network will get compromised.

To establish a pair wire key between nodes, the procedure is:

Step 1: Construct a $(\lambda+1)\times N$ matrix G over a finite field GF (q) where, N represents the number of nodes and q>N. G is a public matrix, so even adversary may know about this matrix. Matrix G is a Vandermonde matrix which takes the forms as follows:

$$G = \begin{bmatrix} 1 & 1 & 1 & . & . & 1 \\ s & s^2 & s^3 & . & . & s^N \\ s^2 & (s^2)^2 & (s^3)^2 & . & . & (s^N)^2 \\ . & . & . & . & . & . \\ s^\lambda & (s^2)^\lambda & (s^3)^\lambda & . & . & (s^N)^\lambda \end{bmatrix}$$

The use of Vandermonde matrix is, instead of storing an entire column the seed can be stored and from that each node can compute the column values.

Step 2: Construct the another matrix $(\lambda + 1) \times (\lambda + 1)$ symmetric matrix D over a finite field GF(q). From the two matrix G and D, construct A matrix, ie., $A = (D.G)^T$. Matrix D has to be secret and it should not be disclosed to adversary. All the matrix construction will be done in base station. Since D matrix is symmetric:

Then $AG = (D.G)^T. G = D^T. G^T. G = G^T. D^T. G = G^T. D.G = (A.G)^T$

If $K = A.G$ then $K_{ij} = K_{ji}$. Based on this idea, the pair wise key can be computed as follows.

Consider Node I, which store the i^{th} row of A matrix and seed to generate the column. Similarly Node J stores the j^{th} row of A Matrix and seed. If Node I and J wants to establish a pair wise key K_{ij} and K_{ji} Node I will send its seed or ID to Node j and vice versa. Simultaneously both nodes will compute the pair wise key. As mentioned earlier, Blom scheme is a λ secure properly i.e., if λ nodes get compromised the entire network will get revealed thus resilience is not optimal.

Multi space key distribution scheme: Based on Blom scheme Du *et al.* (2005) proposed a scheme (Marcos *et al*., 2010) with small changes. Blom Scheme is a perfect pair wise scheme every node will communicate with every other node i.e. It is a complete graph. But this is not necessary in case of WSN. Any two nodes can communicate i.e. it can be connected graph rather than complete graph. As like Blom scheme, matrixes are constructed i.e. public matrix G and symmetric matrix D. But in this case instead of one key space multiple key spaces i.e. multiple D Matrixes like D_1, D_2,.., D_ω are constructed. Blom scheme is a single key space but. Du *et al.* (2005) uses multiple key spaces. Various phases involved in Du *et al*. (2005) scheme to establish a pair wise scheme is In Key Pre-Distribution Phase: All nodes will have unique identify ranging from 1to N. Generate G matrix and Generate ω key spaces. Since Du *et al*. (2005) scheme lies multiple key spaces, generate ω symmetric matrices D_1, D_2, … D_ω, of size $(\lambda+1)$ $(\lambda+1)$. From the two matrixes G and D matrices compute $A_i = (D_i . G)^T$ and select Γ unique key spaces and stored in each node. Nodes which share the same key space will establish a common shared key as Blom scheme. In Key agreement phase, after deployment each node has to establish a shared key with its neighbor for that each node broadcast its node identifier and index of the key spaces. The nodes which do not share common key spaces have to undergo path key establishment phase as in EG Scheme (Eschenauer and Gligor, 2002). Since Γ key spaces are loaded into sensor node and each key spaces is having $(\lambda + 1)$ elements, the total memory usage will be $M = (\lambda + 1) \Gamma$. This scheme achieves optimal resilience but incurs computation overhead as like Blom's scheme. Scalability is another issue. To achieve key connectivity this scheme undergo path key establishment phase which incurs communication overhead.

LU matrix: In LU matrix (Dai *et al*., 2010), two matrixes called Lower triangular and Upper triangular matrix are involved. Both the matrix is used to generate the secret Key matrix for sensor nodes. From the randomly generated key pool, Lower triangular matrix is generated. Based on assumptions that product of L matrix and U matrix will produce symmetric matrix K, the Upper triangular matrix is generated. The relationship between L matrix, U matrix and K matrix is clearly given as follows:

$$\begin{bmatrix} l11 & o & 0 \\ l21 & l22 & 0 \\ l31 & l32 & l33 \end{bmatrix} . \begin{bmatrix} u11 & u12 & u13 \\ 0 & u22 & u23 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} k11 & k12 & k13 \\ k21 & k22 & k23 \\ k31 & k32 & k33 \end{bmatrix}$$

Since K matrix is symmetric, K_{ij} will be equal to K_{ji}, the value of U_{ij} is calculated using the following relationships and these are:

$k_{12} = k_{21} = l_{11}.u_{12} = l_{21}.u_{11} \Rightarrow u_{12} = (l_{21}.u_{11}) / l_{11}$
$k_{13} = k_{31} = l_{11}.u_{13} = l_{31}.u_{11} \Rightarrow u_{13} = (l_{31}.u_{11}) / l_{11}$
$k_{23} = k_{32} = l_{21}.u_{13} + l_{22}.u_{23} = l_{31}.u_{12} + l_{32}.u_{22}$
$u_{23} = (l_{31}.u_{12} + l_{32}.u_{22} - l_{21}.u_{13}) / l_{22}$

Using the random values of u_{11}, u_{22} and u_{33}, the remaining unknown values of u matrix are calculated. The important condition is the product of L and U matrix would be symmetric.

From the value of L and U matrix, the value of K is calculated:

$l_{11}.u_{11} = k_{11}$; $l_{11}.u_{12} = k_{12}$; $l_{11}.u_{13} = k_{13}$; $l_{21}.u_{11} = k_{21}$; $l_{31}.u_{11} = k_{31}$
$l_{21}.u_{12} + l_{22}.u_{22} = k_{22}$; $l_{21}.u_{13} + l_{22}.u_{23} = k_{23}$; $l_{31}.u_{12} + l_{32}.u_{22} = k_{32}$
$l_{31}.u_{13} + l_{32}.u_{23} + l_{33}.u_{33} = k_{33}$

In general when pair-wise scheme is consider, many have an assumption that it is suitable for DWSN. But the drawback in DWSN every node must communicate with every other node since sensor nodes are scattered and have short transmission range, establishing a pair wise key is difficult. Storage is also not efficient. When pair wise scheme is implemented in HWSN, group of sensor node will form the cluster so the key loaded into sensor node will be less. But the problem is most of the information will be loaded in cluster head. So when cluster Head gets compromised all information will get revealed.

Hierarchical LU: Based on all above mentioned problems Wen *et al*. (2007) proposed a key management scheme using LU matrix for HWSN with small information's loaded in cluster head (Wen *et al*., 2007). This scheme is three tier architecture: base station, cluster head and sensor nodes. Two different symmetric key matrixes are generated: SM_{CH} and $SM_{CS.}$ SM_{CH} is used to communicate between cluster heads. SM_{CS} is used to communicate between cluster head and sensor nodes.

Initially all nodes are loaded with one row and one column from its corresponding LU matrix. Rows and Column are selected randomly. If two nodes or two cluster heads or cluster head and sensor nodes wants to communicate procedure to follow is: (between Cluster Head)

- Cluster head CH_i send its column U_{Chi_ci} to CH_j
- Once after receiving Cluster head CH_j. compute K_{CH_ji} and send its column and $F(K_{CH\text{-}ji})$ to CH_i
- Cluster Head CH_i compute K_{CH_ij} and verifies $F(K_{CH\_ij}) = F(K_{CH\_ji})$. If equals it sends {ok, $F(K_{CH\_ij})$} else it sends error message which will be authenticated by the receiver using any authentication methods such as TESLA (Perrig *et al*., 2002)

The Key K_{CH_ji} will be used as pair wise key between two cluster heads.

In this scheme when cluster head get compromised nothing will get revealed, since all information is stored in base station. But some node has to be act as cluster head because sensor nodes are not in contact with base station. Any node cannot be elected as cluster head also because that node doesn't have information about SM_{CH}. The change to be done is replace with other cluster head. When compared with the blom scheme (Blom, 1985), transmission overhead is more because blom scheme broadcast the seed of column but in this scheme the entire column will be broadcasted, but computational complexity is less because no need to generate the column. In LU matrix most of the row and column elements will be zero, so memory requirement is less. But the problem with this scheme is scalability because when more number of sensor nodes is added to the network possibility of selecting the same rows and columns from the matrix will be high.

Based on this hierarchical based LU matrix scheme, (Manivannan *et al*., 2011) proposed another scheme will some added advantage. In this scheme both node to node communication and group communication has been achieved using LU matrix. But group communication is used for commanding the node rather than sending data with the group key. Because in the above scheme if any node get compromised it has to be unicasted to each node by cluster head. So communication overhead is more. The methodology behind this scheme for group communication is the same row and same column is loaded into group of sensors. So all nodes will share a same key, for group communication. Regarding the scalability, when number of nodes added is more, instead of selecting row and column from the same LU matrix, base station will create another new LU matrix and load row and column from that matrix, so repetition of keys will be avoided.

POLYNOMIAL BASED SCHEMES

Blundo's scheme: This scheme (Blundo *et al*., 1993) is based on polynomial based key pre distribution scheme. Blundo scheme uses λ-degree bivariate polynomial $F(x, y) = \sum_{i}^{\lambda} j = 0\, a_{ij}x^i y^j$ over GF (q) which satisfies the property f(x, y) = f(y,x),q is a prime and cryptographic keys should be within this q. Once generating the polynomial, polynomial shares are loaded into sensor node. For example node i get it polynomial share as f (I, y). Similarly node j gets its polynomial share as f (j, y). To establish common key between two nodes i, j, node i evaluate f (i, y) at node j and node j evaluate f (j, y) at node i. Memory requirements will be $(\lambda+1)\log_2 q$. Like Blom's scheme (Blom, 1985) it won't cause communication overhead but Blundo scheme also λ secure property. If λ nodes get captured, the entire network information will get revealed.

Polynomial pool-based key pre distribution: Based on Blundo's scheme (Blundo *et al*., 1993) and basic scheme Liu et al proposed a polynomial pool based scheme (Liu *et al*., 2003) similar to Du *et al*. (2005) multiple space key pre distribution scheme. Generate ω randomly bivariate polynomials of the form $f(x, y) = \sum_{i}^{\lambda} j = 0\, a_{ij}x^i y^j$ called polynomial pools. As basic scheme, randomly select the polynomial and load into sensor node Liu *et al*. proposed

two instances establish a communication key. In the first approach i.e., random subset assignment each node is loaded with set of id's with which the node will share the common polynomial. Though these schemes simplify the shared key discover phase and achieves perfect resilience, scalability will be an issue. To overcome this basic scheme methodology can also be used i.e.; randomly select the polynomial and load in to sensor nodes. In shared key discover phase nodes will broad cast its polynomial ids. The node which shares the common polynomial id will establish a common key. The nodes which do not establish a common key will undergo path key establishment phase. Problems with this scheme is that if any one communication link is compromised , chance for compromising remaininglinks also possible since polynomials are randomly drawn from polynomial pool, So chance for repetition is possible.

Second approach is based on grid based pre-distribution in which m × m 2 dimensional grid is constructed from a set of 2m polynomial i.e., $f^{c}_{\alpha}(x, y)$ $f^{r}_{b}(x, y)$ where, $1 \leq \alpha, ß \leq m$, $m = (\sqrt{n})$, where, n represent the size of the network. Each row α is associated with the polynomial $f^{r}_{b}(x, y)$ and each column β is associated with the polynomial $f^{c}_{b}(x, y)$. In the grid, sensor node i is assigned to (α, β) and the polynomial share is $f^{c}_{\alpha}(x, y)$ and $f^{R}_{B}(x, y)$. During shared key discovery phase node i and j will establish a key if $c_i = c_j$ or $r_i = r_j$. If not equal path key establishment phase is performed. This grid based scheme is further extended to hyper cube, instead of 2 dimensions Γ dimensions grid is constructed. If the Γ is high, key connectivity will be less during shared key discovery phase. The advantage of using grid based and hypercube based scheme is that the id's are not broadcasted so communication overhead will be less and achieves optimal resilience.

DEPLOYMENT KNOWLEDGE

Extending probabilistic schemes: The schemes which proposed earlier does not depends upon deployment knowledge since sensor nodes are randomly scattered in an area, it is hard to find deployment region. But sometimes for example if sensor nodes are scattered to particular point, those sensor nodes can be grouped in particular pattern and based on this node location can be obtained. If deployment knowledge is incorporated in key management scheme it leads to better storage, connectivity and better resilience. Du *et al*. (2005) first proposed deployment knowledge based Key management (Du *et al*., 2004) scheme which is the extension of basic scheme. This scheme has undergone two models:

- Node deployment knowledge in the network
- Develop a key pre-distribution scheme

Node deployment knowledge in the network: If N sensor nodes are deployed in a location, they are divided into equal size groups and grid base approach is used for deployment mode. Groups which deployed closely will share common keys. If the distance between the groups increases the overlapping of keys will get decreased. In basic scheme keys are loaded into sensor nodes from the same key pool but in this scheme keys are loaded from different pools for different group. Ultimately the key pool |S| is divided into sub key pool. Each pool will have |Sc| Keys.

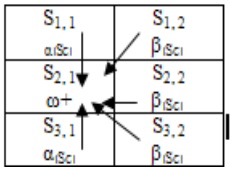


Fig. 1: Selecting keys for key pool $S_{2,1}$

Key pre-distribution scheme: This scheme also consists of three phase: Key pre-distribution, Shared Key discovery and path key establishment. First phase alone is different in this scheme, rest of the phases are same like EG scheme (Eschenauer and Gligor, 2002). In key pre-distribution phase, key pool is divided into t x n key pools. While dividing the key pool, the neighboring pools will have more common keys. Each node in the group is loaded with the key from corresponding key pool.

To create key pools: Since deployment model follows grid based approach:

- Pools which are horizontal or vertical share $\alpha|S_c|$ keys, where, $0 \leq \alpha \leq 2.5$
- Key pools which are diagonal share $\beta|S_c|$ keys, where $0 \leq \beta \leq 2.5$
- Two non neighboring pools don't share keys.

The procedure for selecting the key for various key pools is

- Select $|S_c|$ keys from |S| for the group placed in $S_{1,1}$ and remove those key from |S|.
- Select $\alpha |S_c|$ for $S_{1,2}$ from $S_{1,2}$ and the remaining $\omega = (1-\alpha) |S_c|$ keys from key pool |S| an remove the selected ω keys from |S|.
- Select $\alpha|S_c|$ keys for $S_{2,1}$ from $S_{1,2}$, $S_{2,2}$, $S_{3,1}$ and select $\beta |S_c|$ from $S_{1,2}$ and $S_{3,2}$. Finally select and remove ω keys from key pool |S|.

Figure 1 Shows that, how to select the keys for the key pool $S_{2,1}$.

To achieve perfect resilience same key should not be shared between the groups, for that also rules should be established i.e., if group G1 select keys from G2, no other

group should not select the same key. Similarly the overlapping factor α and β should be assumed properly. For example α = 0.25 and β = 0; only horizontal and vertical pools will share the key. Performance analysis of this scheme shows that good in storage, connectivity and resilience factors. But the disadvantage is its complexity.

Extending matrix based schemes: Yu and Guan (2008) proposed deployment knowledge scheme with Blom's (1985) scheme where N sensors nodes are divided into groups. As Blom scheme each sensor in the group is loaded with seed to generate the column of G matrix and one row from D_i matrix so nodes within the group will establish a key. To provide inter group communication each node also store t rows from D_i matrix such that at least there will be one common D_i matrix between the neighbors to establish a key. The advantage of this scheme is that key connectivity is achieved. As Blom's scheme, computation overhead is more i.e. to generate the column G using seed and multiplication to get the key. Similarly storage is t x (λ+1) elements where t represents the number of rows of D_i matrix, λ+1 represent the elements in each column of D_i matrix.

Group keying: In group keying, there will be clusters called group. Each group will share common group key for communication. For group keying four parameters to be consider n-number of sensor nodes, k-number of keys loaded into sensor nodes, m-rekeying message and s-session key. In group communication, all nodes will use single session key for communication. All data are encrypted using session which is common to all nodes. So this allow data aggregation and sometimes dissemination if duplication. Similarly the value of k and m should depends upon the situation i.e. if network size is large m should be increased because the number of keys stored will be increased and if the sensor eviction rate is high, the N should be low. k administrative keys are loaded into sensor nodes, which are used for rekeying when nodes get compromised or when new nodes get added. Sometimes individual keys are used for authentication and for initialization when new user joined the group. For group keying, the two approaches widely used are Tree based and EBS.

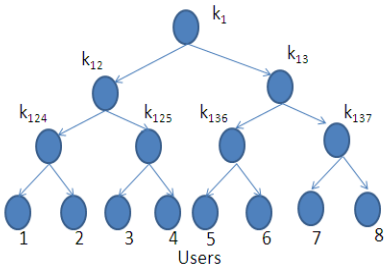


Fig. 2: Binary tree structure

	M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9
K_1	1	1	1	1	1	1	0	0	0	0
K_2	1	1	1	0	0	0	1	1	1	0
K_3	1	0	0	1	1	0	1	1	0	1
K_4	0	1	0	1	0	1	1	0	1	1
K_5	0	0	1	0	1	1	0	1	1	1

Fig. 3: Canonical matrix (5, 3)

Tree based approach: Tree based Approach: In tree based approach (Eltoweissy *et al*., 2004), for k-ary trees the number of keys stored in each node will be the height of the tree i.e. $\log_k n$. For binary tree it is $\log_2 n$ keys. Compared to all trees, binary tree will have less rekeying message. Figure 2 Shows a binary tree structure.

User1 knows {K_1, K_{12} & K_{124}}. If suppose if user 1 evicts, keys K_{124}, K_{12} and K_1 have to be changed. So the number of rekeying messages required will be three. As a result, if the height of the tree get increased i.e. number of user increased, the number of keys stored will also be increased. Simultaneously rekeying message also increased. This leads to communication overhead and storage also an issue.

EBS-exclusion basis system: Eltoweissy *et al*. (2004) proposed a new key management technique called EBS (Exclusion Basis System) (Eltoweissy *et al*., 2004) which gives optimal result for the parameters n, k, m when compared to tree based approach.

Definition: Choose the value of n, k and m such a way that k>1 and n>m. An exclusion basis system denoted by EBS (n, k, m) is a collection of subsets of (1,n) = {1,2…n} such that for every integer t ϵ (1, n) the following two properties hold.

Property (a): t is in at most k subsets of Γ and $U^m i = 1$ i $A_i$is (1,n)-{t}

Property (b): There are exactly m subsets say A_1, A_2…, Am in Γ such that $U^m i = 1$ A_i is (1 , n)-{t}.

Consider an example EBS (10, 3, 2); The canonical matrix for C (5,3) is shown in Fig. 3:

Subsets are ; A_1 = {1,2,3,4,5,6}; A_2 = {1, 2, 3, 7, 8, 9}; A_3 = {1, 4, 5, 7, 8, 10}

A_4 = {2, 4, 6, 7, 9, 10}; A_5 = {3, 5, 6, 8, 9, 10}

According to first property since k = 3 each element t is in at most k subsets for example the element 1 is in {A_1, A_2, A_3} and 10 is in {A_3, A_4, A_5} etc.

According to second property each element is excluded by union two subsets:

$A_1UA_2 = (1, 10)-\{10\}$

$A_1UA_3 = (1, 10)-\{3\}$

$A_4UA_5 = (1, 10)-\{1\}$

In the above example, since k is 3, each user will have 3 administrative keys i.e., user 1 will have $\{k_1, K_2, K_3\}$, for user 2 $\{K_1, K_2, K_4\}$, etc.

Use of administrative keys: As explained earlier administrative keys are used for rekeying i.e, when user gets compromised/added.

For example when user 1 is evicted in EBS (10, 3, 2) the steps to be taken are: User 1 knows the key $\{K_1, K_2, K_3\}$, so these keys have to be changed.

The steps are Generate the new key K_1', K_2', K_3' and encrypted by former key K_1, K_2, K_3.

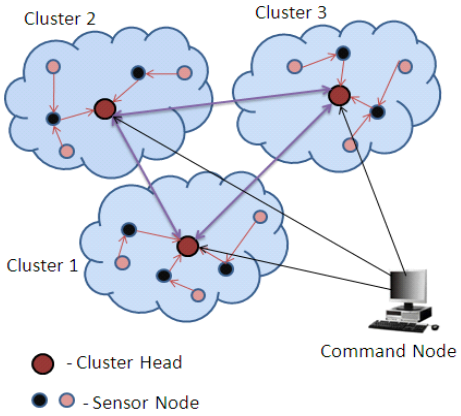
Generate the new session key S', since the rekeying message m = 2 by sending two message the previous key can be changed. i.e., K_4 (S', $K_1(K_1')$, $K_2(K_2')$, $K_3(K_3')$,) & K_5 (S', $K_1(K_1')$, $K_2(K_2')$, $K_3(K_3')$,)

Construction of EBS: To construct EBS (n,k,m), canonical enumeration has to be found i.e., combination of forming subsets of K objects from a set of k+m objects. Positive solutions to EBS (n, k, m) can be obtained if and only if C(k+m, k) ≥n. Though the number of keys stored in binary tree is greater than the EBS (Eltoweissy *et al*., 2004), collision is not possible in binary tree. This is the major drawback in EBS. For example in EBS (10, 3, 2). User 1 and user 6 get compromised the entire key set will get revealed.

SHELL: Based on this, Younis *et al*. (2006) proposed a location aware combinatorial Key management scheme for clustered sensor nodes called SHELL (Younis *et al*., 2006), Which incorporates the features of EBS (Eltoweissy *et al*., 2004). Shell Exclusion base system has been designed for secure group communication for wired and adhoc networks. SHELL is a collision resistant scheme. The major components involved in SHELL are command node, Gateway, Key generating Gateways and sensor nodes. Command node or Base station communicates with gateways and is capable to detect compromised gateway. Gateway can communicate with other Gateway and also with the sensor node within its cluster. Gateway is capable to detect compromised sensor node. Gateway will form the EBS matrix, but actual keys will be generated by KGN. This is because if gateway gets compromised nothing will get revealed. In a cluster, there will be more than two KGN. The administrative keys will be loaded into sensor node from all KGN. The major problem in EBS Scheme is collision as explained above.

KC	1	2	3	4	5	6	7	8	9	10
1	0	2	2	2	2	2	2	4	4	4
2	2	0	2	2	2	4	4	2	2	4
3	2	2	0	2	4	2	4	2	4	2
4	2	2	2	0	4	4	2	4	2	2
5	2	2	4	4	0	2	2	2	2	4
6	4	2	4	2	2	0	2	2	4	2
7	2	4	4	2	2	2	0	4	2	2
8	4	2	2	4	2	2	4	0	2	2
9	4	2	4	2	2	4	2	2	0	2
10	4	4	2	2	4	2	2	2	2	0

Fig. 4: Hamming distance



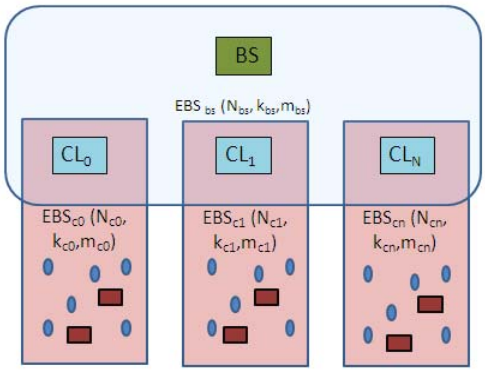


Fig. 5: SHELL Vs LOCK

Collision prevention: Since SHELL is based on location aware; the key collision can be avoided. While constructing EBS table, Canonical enumeration is formed as in Fig. 3.

Each column will have 3 1's i.e., each node will know 3 administrative keys. To prevent collision hamming distance has to be found out i.e. the number of bits that the combinations of two nodes differ as shown in Fig. 4. The value of d will be:

$2 \le d \le 2k \quad k<m$
$2 \le d \le 2\,m \quad m<k$
$2 \le d \le k+m \quad k = m$

In general if the hamming distance is less, the keys revealed will be less. Therefore before deploying the sensor node, Hamming Distance has to be calculated. But it leads to complexity.

LOCK: Based on EBS, Etoweissy *et al*. (2004) proposed a dynamic key management scheme called Localized Combinatorial Keying (LOCK) (Eltoweissy *et al*., 2004). In this scheme administrative keys and session keys have to be changed periodically to achieve resilience, failures and topology changes. Though SHELL (Younis *et al*., 2006) is collision resistant rekeying is fully depend on centralized key server and also location dependent. But LOCK is post deployment i.e., information about the location of nodes are not required. Architecture of LOCK is similar to SHELL i.e., three tiers with common node, cluster head and sensor nodes. But the main difference is communication mechanism between command node and cluster head as shown in Fig. 5. Two layers of EBS (Eltoweissy *et al*., 2004) keys are constructed. First layer between command node and cluster head and second layer between cluster head and sensor nodes. But in SHELL, pair-wise scheme is incorporated between command node and cluster head. During Initialization phase some backup keys are loaded into sensor nodes for authentication when cluster head get compromised. SHELL fails to provide authentication. When comparing LOCK with static schemes, the number of keys loaded into sensor nodes is less. The use of key polynomials in LOCK is to achieve resilience avoid location based information.

LEAP: LEAP (Localized Encryption and Authentication Protocol), protocol is designed for heterogeneous networks. LEAP (Zhu *et al*., 2003) support all types of communication pattern i.e. Network, Pair-wise and Group keying. In addition to that it supports the featureslike in network processing, passive participation and data fusion. In WSN since sensor nodes are closely deployed, possibility of sensing the same information will be more. If same value is send by an individual node, the communication cost and redundancy will be more. So network life time is reduced. To avoid these problems LEAP (Zhu *et al*., 2003) come out with the features of data fusion. i.e. it will gather the information then in-networks processing. It means processing on gathered information before transmitting. It support basic requirements such as confidentiality, authentication and it avoids attacks like warm-hole and Sybil. For communication pattern, LEAP uses four different keys. These are Individual key, pair-wise shared key, cluster key and group key.

Individual key: Individual Key K_{BS} is used to communicate between base station and sensor nodes. This key is loaded into sensor node before deployment. The base station will not store all the individual keys of the node. When base station wants to communicate with sensor node, it generates the key using Pseudorandom function f and its master key i.e., $K_{BS} = f(K_B)$.

Pair-wise shared key: This key is shared between node and its neighboring node. It is used for sending the cluster key and data to the nearby data fusion node. Sensor node will establish a pair-wise key with neighboring nodes after deployment. To establish a pair-wise key, the four stages are followed. These are key pre-distribution, neighbor discovery, pair-wise key establishment and key erasure.

Initially all sensor nodes are loaded with key K_i and from that node will derive the master key K_u. If node U wants to discover its neighbors, first it will initialize the timer because the sender node has to identify its neighbor before t_{min}; else the attacker will compromise the network. Then node U broadcast HELLO message with its identity.

Node U computes the pair-wise key K_{UV} using node V's identity. Node V will also follow the same procedure. In the fourth phase node U erase k1 and all the keys of neighboring node after t minimum.

Cluster key: As LEAP (Zhu *et al*., 2003) support data aggregation, some node will have the responsibility to decide while forwarding the data to avoid duplication and thus communication overhead can be reduced. For that neighboring node will share the same key which is called cluster key, this will be distributed using pair-wise key.

Group key: Base station shares this key with all nodes in the network to send commands. This key either pre loaded or it can be distributed using cluster key. Proper rekeying mechanism should be handled while changing the group key.

The advantage of LEAP is support all four types of communication pattern, provide authentication and avoid attacks like Sybil, Sink hole etc. Because of it's, in-network processing, network lifetime can be increased. The disadvantage of this scheme is computation and communication overhead increases when the density of the network increases and storage also an issue.

DISCUSSION

While designing KMS for WSN the metric to be evaluated against KMS are key connectivity, resilience, efficiency and scalability. In general achieving all metrics in a single key management scheme is difficult. Based on keying mechanism, metrics can be evaluated i.e. for pair-wise scheme resilience and key connectivity and for group keying scalability can be measured. Efficiency plays a vital role for both schemes. Other than that other metrics like deployment knowledge, node authentication can also be considered. Table 1 shows the summary of all

Table 1: Summary: Key management schemes

Schemes		Key connectivity (%)	Authenti cation	Scalability	Resilience-resistance against node capture	Deployment knowledge
Full pair-wise	100	Yes	Difficult	100%	No	
	E-G scheme	Achieve 100 after path key establishment phase	No	Difficult	K/ |P|	No
Probabilistic	Q composite	Achieve 100 after path key establishment phase	No	Difficult	k |P|q q	No
	Multipath Key Pre-distribution	Achieve 100 after path key establishment phase	No	Difficult	100%	No
	Random pair -wise	100	Yes	Difficult upto (n-m) nodes	100%No	
	AP scheme	Achieve 100 after path key establishment phase	No	fair	100% since H sensor are tamper resistant	No
	Blom's	100	Yes	Difficult	λ Secure	No
	Du *et al.*	100 but Complexity	Yes	Difficult	λ Secure	No
	LU hierarchical	100	No	Difficult	λ Secure	No
Matrix based	LU hierarchical	100	No	Difficult	λ Secure	No
	LU (19)	100	Yes	Good	λ Secure	Yes
Polynomial		100	Yes	Good	λ Secure	No
	EBS	100	No	Good	0 %	Yes
	SHELL	100	No	Good	0%	Yes
	LOCK	100	Yes	Good	0%	No
Group	LEAP	100	Yes	Good	Pair-wise-100% group-0% individual-100 % cluster-0%No	
	Extended basic	Achieve 100 after path key establishment phase	No	Difficult	k / $\|S_c\|$	Yes
Deployment scheme	Extended matrix based	100	Yes	Difficult	λ Secure	Yes
	Extended polynomial	Achieve 100 after path key establishment phase	Yes	Good	λ Secure	Yes

schemes. Each scheme will have both advantage and disadvantage while considering basic scheme though it is simple, efficient and achieve optimal resilience but fails to provide authentication. Basic scheme have been modified as Q-composite scheme to achieve better resilience. Random pair-wise scheme have been proposed to avoid memory usage but the drawback is, it cannot be extended not more than n-m nodes. To achieve better resilience in basic scheme multipath key reinforcement scheme is proposed. In that instead of using single path multi path is used for key redistribution. To avoid path key establishment in basic scheme, key redistribution scheme have been proposed. All the above methodology concentrates on key connectivity and resilience. Node revocation is also another issue in the above scheme; which will be handled by the controller node. Node authentication is also an issue. In matrix based scheme there will be 100% key connectivity. It provides node authentication also. But the problem with this scheme is scalability. If deployment knowledge is incorporated in matrix based scheme, it can be implemented for group communication to achieve inter and intra cluster communication. But matrix schemes are λ secure properties i.e., if λ nodes get compromised the entire network will get revealed. Polynomial pool based schemes achieves scalability but also λ secure property. All above mentioned schemes are perfect for node to node communication. But Blundo scheme is used for secure group communication also.

For group communication, Exclusion Basis Scheme has been designed. But in EBS collision will occurs. To avoid collision, SHELL has been proposed but it must need deployment knowledge. SHELL fails in authentication. Based on SHELL, LOCK has been proposed but this scheme is post deployment and provides authentication. LEAP supports both node to node communication and group communication. But storage is an issue because it has to store large key for achieving all communication pattern. The comparison is made between different key management scheme in distributed and hierarchical. The Table 1 shows summarize the list of DWSN and HWSN.

CONCLUSION

Security plays a vital role in wireless sensor networks. Key management is an essential security issue

Table 2: DWSN Vs HWSN

Distributed WSN	Hierarchical WSN
EG Scheme, Q Composite,	AP Scheme
Multipath Key Reinforcement Scheme,	LU Matrix and hierarchical LU matrix
Random Pair-wise Key Scheme,	EBS: SHELL, LOCK and LEAP.
Key redistribution scheme.	Group Keying: Tree based approach,
Matrix Based: Blom's scheme, multi space key distribution scheme	
Polynomial KMS: Blundo, Polynomial pool based key pre-distribution scheme	
Deployment Knowledge: Extending Probabilistic scheme, Extending matrix based scheme	

in wireless sensor networks. Implementing key management schemes along with cryptographic algorithms are complex because of sensor node constraints. In this paper a review of key management schemes both in Distributed wireless sensor networks and Hierarchical wireless sensor networks have been done based on keying mechanisms, such as network, pairwise and group keying. In various key management schemes different evaluation metrics are discussed and it has been summarized in Table 2. Finally from key management scheme are concentrating on some evaluation metrics rather than all.

REFERENCES

Blom, R., 1985. An Optimal Class of Symmetric Key Generation Systems. Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, Springer-Verlag Berlin Heidelberg, pp: 335-338.

Blundo, C., A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, 1993. Perfectly Secure Key Ddistribution for Dynamic Conferences. In: LNCS, Springer-Verlag Berlin Heidelberg, 740: 471-486.

Çamtepe, S. and B. Yener, 2005. Key Distribution Mechanisms for Wireless Sensor Networks: A Survey. In Technical report, Rensselaer Polytechnic Institute, TR-05-07.

Chan, H., A. Perrig and D. Song, 2003. Random Key Predistribution Schemes for Sensor Networks. In: Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP'03), IEEE Computer Society, Washington, DC, USA, pp: 197-213.

Dai, H. and H. Xu, 2010. Key predistribution approach in wireless sensor networks using LU matrix. IEEE Sens. J., 10(8): 1399-1409.

Du, W., J. Deng, Y. Han, S. Chen and P. Varshney, 2004. A key management scheme for wireless sensor networks using deployment knowledge. IEEE Conference Computer and Communications Societies, IEEE Computer Society, vol. 1, Los Alamitos, CA, USA, pp: 586-597.

Du, W., J. Deng, Y. Han, P. Varshney, J. Katz and A. Khalili, 2005. A pairwise key pre-distribution scheme for wireless sensor networks. ACM T. Inform. Syst. Se., 8(2): 228-258.

Du, X., Y. Xiao, M. Guizani and H.H. Chen, 2007. An effective key management scheme for heterogeneous sensor networks. Ad. Hoc. Networks, Elsevier, 5(1): 24-34.

Eltoweissy, M., H. Heydari, L. Morales and H. Sadborough, 2004. Combinatorial optimization of key management in group communications. Network Syst. Manage., 12(1): 33-50.

Eschenauer, L. and V. Gligor, 2002. A Key-Management Scheme for Distributed Sensor Networks. In ACM Conference on Computer and Communications Security (CCS'02), ACM, New York, USA, pp: 41-47.

Lai, B., S. Kim and I. Verbauwhede, 2002. Scalable Session Key Construction Protocol for Wireless Sensor Networks. In IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES), IEEE Computer Society, Washington, DC, USA.

Law, C.F., K.S. Hung and Y.K. Kwok, 2007. A Novel Key Redistribution Scheme for Wireless Sensor Networks. In: IEEE International Conference on Communications, IEEE Computer Society, Washington, DC, USA, pp: 3437-3442.

Liu, D. and P. Ning, 2003. Establishing Pair-wise Keys in Distributed Sensor Networks. Computer and Communications Security, ACM, New York, USA, pp: 52-61.

Manivannan, D., R. Ezhilarasie and P.A. Neelamegam, 2011. An Efficient and Hybrid Key Management Scheme for Three Tier Wireless Sensor Networks Using LU Matrix. CCIS 192, Springer-Verlag Berlin Heidelberg, pp: 111-121.

Marcos, A.S., P.S.L.M. Barreto, B.M. Cintia and T.C.M.B. Carvalho, 2010. A survey on key management mechanisms for distributed wireless sensor networks. Comput. Network, 54(15): 2591-2612.

Perrig, A., R. Canetti, J.D. Tygar and D. Song, 2002. The Tesla Broadcast Authentication Protocol. RSA Crypto Bytes, pp: 1-13.

Wen, M., Y. Zheng, H. Li and K. Chen, 2007. A Hierarchical Composition of LU matrix-based Key Distribution Scheme for Sensor Networks. Springer-Verlag Berlin, Heidelberg, pp: 608-620.

Yang, X., K.R. Venkata, S. Bo, D. Xiaojiang, H. Fei and G., Michael, 2007. A survey of key management schemes in wireless sensor networks. Comput. Commun., 30(11-12): 2314-2341.

Younis, M.F., K. Ghumman and M.A. Eltoweissy, 2006. Location-aware combinatorial key management scheme for clustered sensor networks. IEEE T. Parallel Distr. Syst., 17(8): 865-882.

Yu, Z. and Y. Guan, 2008. A key management scheme using deployment knowledge for wireless sensor networks. IEEE Trans. Parallel Distr. Syst., 19: 1411-1425.

Zhu, S., S. Setia and S. Jajodia, 2003. LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks. Computer and Communications Security, Washington USA.