Classification of Symmetric Key Management Schemes for Wireless Sensor Networks

Suman Bala, Gaurav Sharma and Anil K. Verma

Thapar University, Patiala, India
suman1005@gmail.com, Gaurav.sharma@thapar.edu, akverma@thapar.edu

Abstract

WSN is the collection of thousands of tiny sensor nodes, which have the capability of sensing, computing and transmitting the information in the network. Due to the low circuit design, it has some resource constraints but efficient to carry the information through wireless communication. But the exchange of information in a secure manner is critical in WSN. There are many techniques developed in recent years for the security purposes, one of the area is the key management. Key management is the challenging issue in sensor networks. In this paper we present the key management techniques for wireless sensor networks and classification has been presented based on the encryption techniques.

Keywords: *Wireless sensor networks, Key Management*

1. Introduction

A WSN is a collection of thousands of resource constrained sensor nodes, which can communicate through wireless medium. These nodes are preferable because they are inexpensive, self-organized and easy to deploy, but due to limited battery, limited processing power, limited memory and wireless nature these are easy to get control over it. Security of WSN is a very important aspect because they carry sensitive information that may be captured by intruder or different types of attack can be played over it. WSN has both military and civilian applications such as detecting and monitoring enemy movement, battlefield surveillance, detection of chemical or biological attack, traffic monitoring, health care and forest fire detection. Due to limited resources in WSN different types of attacks like Denial of Service, node tampering, eavesdropping can be easily implemented. Therefore there should be some flexible and effective mechanisms for secure communication in WSN.

Key management protocols are the backbone for security in WSN. The main goal of key management scheme is to provide secure communication between sensor to sensor, a group of sensor and sensor to base station. These are known as unicast, multicast and broadcast respectively. Key management is a bundle of components such as key establishment- protocol in which shared secret keys are available to both the parties. Several researchers have categorized the key management techniques into three categories: symmetric, asymmetric and hybrid. The paper presents the classification of symmetric key management schemes for WSN.

2. Classification of Symmetric Key Management Schemes

There are various ways in which we can classify key management schemes in wireless sensor networks by considering different benchmarks. Various researchers gave different taxonomies. Key management schemes in wireless sensor networks can be classified broadly into dynamic or static solutions based on whether rekeying of administrative keys is enabled post network deployment. Schemes are also classified into homogeneous or heterogeneous schemes with regard to the role of network nodes in the key management process. Homogeneous schemes generally assume a flat network model, while heterogeneous schemes are intended for both flat and clustered networks. Another criterion is hierarchical and distributed sensor networks based on network models. Other classification criteria include whether nodes are anonymous or have pre-deployment identifiers, and if so, when (pre-, post-deployment, or both), and what deployment knowledge (location, degree of hostility, *etc.*) is imparted to the nodes.

We are presenting taxonomy of symmetric key management schemes. Figure 1 shows the classification of symmetric key management schemes for wireless sensor networks. We classify symmetric key management schemes broadly into three categories, these are: base-station participation scheme, trusted third node based scheme and pre-distribution schemes. Based on the key distribution, key discovery and key establishment in the pre-distribution schemes, we classify these schemes into nine categories, these are: master key based pre-distribution scheme, pairwise key pre-distribution schemes, pure probabilistic key pre-distribution schemes, polynomial based key pre-distribution schemes, matrix based key pre-distribution schemes, tree based key pre-distribution schemes, hierarchical key management scheme, combinatorial design based key pre-distribution schemes, EBS based key pre-distribution schemes. We present the merits and demerits of symmetric key management schemes for wireless sensor networks in Table 1. We also present the evolution of the schemes in Table 2.

2.1. Base Station Participation Scheme

In base station participation schemes, a trusted, secure base station is used as an arbiter to provide link keys to sensor nodes. Each sensor node shares a unique key with a base station, which acts as a key distribution center (KDC). Thus, the scheme is also called centralized key distribution center (KDC) approach. The sensor nodes authenticate themselves to the base station. After that the base station generates a link key and sends it to both parties securely. If two nodes must communicate securely, they can acquire a shared key from the base station, which unicasts the key to each of them. The scheme requires less memory and perfectly controlled node replication, also it is resilient to node capture and possible to revoke key pairs. But it is not scalable and the base station becomes the target of attacks.

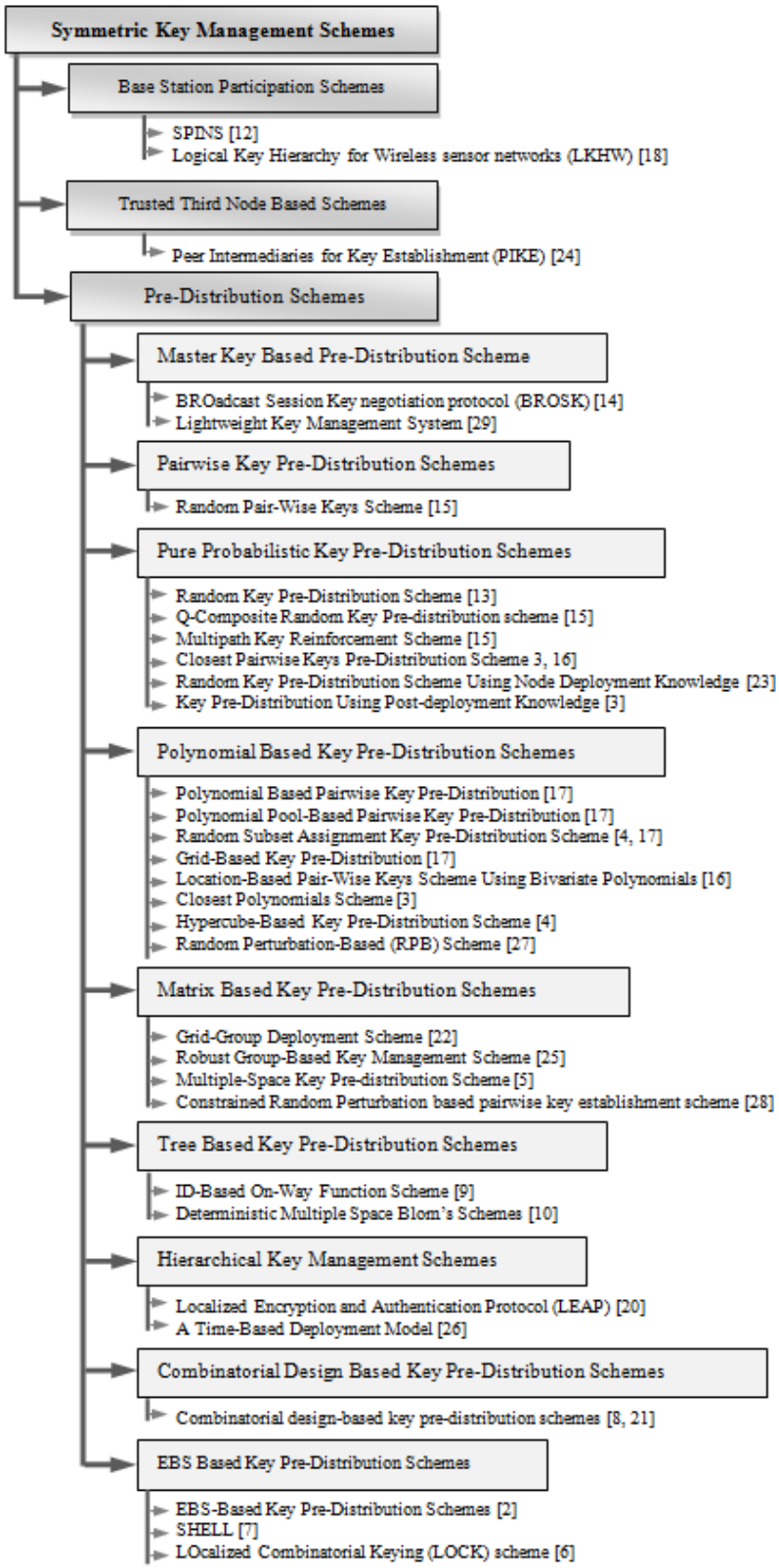

Figure 1. Classification of Symmetric Key Management Schemes

SPINS [12]: SPINS is a security building block that is optimized for resource-constrained environments and wireless communication. It is based on trusted base server. The authors proposed and implemented an authenticated routing scheme and asecure node-to-node key agreement protocol. SPINS is based on two secure building blocks: SNEP (Secure Network Encryption Protocol) and μTESLA (the "micro" version of the Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol). SNEP provides the data confidentiality, two-party data authentication, and data freshness with low overhead. While μTESLA provides authenticated broadcast for severely resource-constrained environments. The protocol is based on some assumptions: (i) minimal hardware infrastructure; (ii) the base station has the capabilities similar to the network nodes, except that it has enough battery power to surpass the lifetime of all sensor nodes, sufficient memory to store cryptographic keys, and means for communicating with outside networks; (iii) all sensor nodes intimately trust the base station: at creation time, each node is given a master key which is shared with the base station, all other keys are derived from the shared key. Thus, each node trusts itself and its sensors. For the design guidelines they use purely symmetric cryptographic primitives to construct the SPINS protocols; and due to limited program store, they constructed all cryptographic primitives like encryption, message authentication code (MAC), hash, random number generator) out of a single block cipher for code reuse and to reduce communication overhead they exploited common state between the communicating parties. The implementation has been done on sensor nodes under the SmartDust program at the EECS buildings, Cory Hall, UC Berkeley. They have evaluated the implementation of their protocols in terms of code size, RAM size, and processor and communication overheads. The result showed that the communication costs are small. The scheme is resilient to node capture and possible to revoke key pairs. But it is not scalable and the base station becomes the target of attacks. SPINS do not provide a solution for denial of service (DoS) attacks when the malicious node keeps sending the request to negotiate a session key because one adversary can easily trigger a REPLAY attack and exhaust the energy in the sensor nodes.

Logical Key Hierarchy for Wireless sensor networks (LKHW) [18]: The Logical Key Hierarchy for Wireless sensor networks (LKHW) is a secure group communication scheme based on directed diffusion and Logical Key Hierarchy (LKH) scheme, to protect the directed diffusion protocol. An LKH is a key tree structure with source nodes as leafs and a sink node as the root. Each leaf node holds keys along the path from it to the root node. In LKHW, an LKH is established before data are fused. Then the LKH is used to provide encryption and authentication for data fusion. The scheme integrates security and routing in a single framework, by leveraging secure multicast techniques and the tried and tested concepts of directed diffusion. The protocol is described as follows: (1) the key distribution center (KDC) sends out "interest about interest to join". (2) The interested nodes reply with "interest to join". (3) The KDC supplies key set and then secure interest and data encrypted with the group key. There are two protocols for the leave and join for dynamic groups. When a node applies to join, a join 'interest' is generated which travels down the gradient that have previously established by 'interest about interest to join'. When a node joins, a key set is generated for the new node based on keys within the key hierarchy. The similar process for the leave is also given.

2.2. Trusted Third Node Based Scheme

In trusted third node based scheme a peer sensor node is used as a trusted intermediary for the establishment of a shared key between nodes.

PIKE [24]: A scheme was proposed known as Peer Intermediaries for Key Establishment (PIKE), a key distribution scheme. A scheme is based on using peer sensor nodes as trusted intermediaries. The scheme was designed to address the lack of scalability of existing symmetric key distribution schemes. The communication overhead of existing schemes or memory per node is $W(n)$. But PIKE achieves a trade-off $O(\sqrt{n})$ overhead in both communications per node and memory per node. The scheme establishes keys between any two nodes regardless of network topology or node density. The scheme is having less communication overhead as compared to KDC approaches. The scheme is much more resilient than the existing one. The scheme has two variations, one is basic PIKE and three or more dimensional PIKE.

The authors proposed a basic PIKE scheme that uses a 2-D grid node ID's to perform a key establishment using at most one intermediary. The scheme uses a concept in which the keys are deployed such that for any two nodes *A* and *B*, it is possible to find some node *C* in the network that shares a unique pairwise scheme with both *A* and *B*. Then *A* can securely route the key establishment message through *C* to *B*. each node in this scheme has an ID of the form *(x, y)*, where $x, y \in \left\{1,2,3,...\sqrt{n}-1\right\}$. Each node *(x, y)* is then loaded with the secret key pair-wise shared solely with each node in the two sets of the nodes:

$$(i,y)\ \forall\ i \in \left\{0,1,2,3,...\sqrt{n}-1\right\} \text{ and } (x,j)\ \forall\ j \in \left\{0,1,2,3,...\sqrt{n}-1\right\}$$

For three dimensional PIKE, each node ID consists of a triple (a1; a2; a3). Each node shares keys with each other node that lies on the same axis as itself, i.e. (i; a_2; a_3), (a_1; i; a_3), and (a_1; a_2; i) for $i \in \left\{0,...,\sqrt[3]{n}\right\}$. Every pair of nodes needs to route their key establishment through at most two intermediaries, or sometimes just one intermediary if both nodes lie on the same plane. Node deployment likewise occurs in node ID order, ensuring that a roughly cuboid area is enclosed by the space of the deployed nodes. The trade-off induced by raising the dimensionality of the node ID space is a reduction of the memory overhead from $O\left(\sqrt{n}\right)$ to $O\left(\sqrt[3]{n}\right)$, while incurring an additional communication and security cost by requiring two intermediaries per key establishment instead of one.

2.3. Pre-Distribution Schemes

In the initialization of the schemes, each node is distributed with the secret keys or secret information before deployed into the sensing area. The key pre-distribution scheme comprises three phases: key pre-distribution phase, shared-key discovery phase and path-key establishment phase.

2.3.1. Master Key Based Pre-Distribution Scheme: A single key is preloaded into all the sensor nodes of the network. After deployment, every node in the network can use this key to encrypt and decrypt messages. The scheme includes minimal storage requirements and avoidance of complex protocols. Since single key is stored in each sensor node; no need for a sensor node to perform key discovery or key exchange. But compromise of a single node causes the compromise of the entire network through the singled shared key.

BROadcast Session Key negotiation protocol (BROSK) [14]: The scheme is based on single master key that is pre-deployed to sensor nodes. They proved that the scheme has better scalability than the existing schemes like SPIN and SNAKE. The scheme is constructing shared secret key where each node can negotiate a session key with its neighbors by broadcasting key negotiation messages. They called shared secret key as shared session key or link-dependent key because different links would use different shared-secret keys. The scheme solves the problem of authentication by constructing trust levels among the nodes. At the initial stage, nodes are having an equal low trust level between them, and the trust will rise by time as the authenticated communication among nodes. The scheme could perform the key negotiating process efficiently as it uses a fully ad-hoc scheme to negotiate the session key. And the scheme is significant for the large-scale sensor networks. The scheme is based on some assumptions (i) Nodes are resource constrained; (ii) Nodes are static or have a low mobility; (iii) Nodes share a master key. The scheme also renegotiates the new session keys; the reason is that it is insecure to use the same session keys for data transmission as the sensor network is working for a while. The simulation has been conducted on a sensor network simulator developed by NESL at UCLA. The result showed that the scalability of BROSK is better than two other security protocols of sensor network, SPINS and SNAKE; the reason is time needed to finish the key negotiation process depends only on the average number of neighbors rather than the total number of nodes. The power consumption is also better, as, SPINS needs four data transmissions to finish the key negotiation process, while SNAKE needs three data transmissions, but in BROSK each node only needs to broadcast once in order to finish the key negotiation process. Thus, the scheme is not having any DoS attack and will not response to false request for key negotiation generated by a malicious node. The schemes require minimal storage and avoidance of complex protocols, hence requires less computations. So, after the deployment, there is no need to perform key discovery or key exchange. The scheme has very low resilience. A single node causes the compromise of the entire network through the shared singled key.

Lightweight Key Management System [29]: The approach requires the sensors to share a small set of secret keys. These keys are loaded in each sensor before deployment, and, unlike other key pre-distribution schemes, the number of keys required does not increase with the network size. Each sensor node stores a group authentication key and a key-generation key. If two sensor nodes are from the same generation, they authenticate each other by using the authentication key. They exchange random nonce values, and establish the session key. They called the process of establishing these pairwise keys bootstrapping and called the corresponding links secure local links. Thus, an extension of the basic bootstrapping protocol is provided by them, which support multiphase deployment, in which secure links are established between sensors that are deployed in different phases. Because of its low cost, this approach is well suited for key management in networks of resource-constrained sensors. The main benefits of the approach can be summarized as follows: (1) Low memory and computation cost: Each sensor needs to store only a small set of (symmetric) keys, independent of network size, and no expensive operations such as those used in public-key cryptography are required; (ii) Low key setup overhead: Sensors deployed at the same time are preconfigured with the same set of keys. As a result, the approach has a small administrative key setup overhead; (iii) Self-organizing: Sensors autonomously establish secure links without involving a trusted server that may become a bottleneck or a single point of failure. Resilience of the scheme is still low because an adversary only needs to compromise the authentication key and a key-generation key of that generation to compromise all the links of nodes in that generation. Adversary may log the messages

flowing in the network to process later when the required credentials are compromised completely.

2.3.2. Pair-Wise Key Pre-Distribution Scheme: In pair-wise key distribution schemes, pairwise keys are loaded to the sensor nodes before deployment. For this, there is a trivial solution: for the *n* sensor nodes in the network, assigning each node a unique pairwise key with all the other nodes in the network, i.e., *n*-1 pairwise keys. This allows each node to communicate with all the nodes in its communication range. This offers node-to-node authentication, increased resilience against the node capture. Thus minimizes the chance for node replication. But the drawback is the additional overhead needed for each node to establish *n*-1 unique key with all the other nodes in the network and maintain those keys in its memory.

Random Pair-Wise Keys Scheme [15]: The random pair-wise keys scheme is a pair-wise key pre-distribution scheme in which distinct pair-wise keys are loaded to the sensor nodes before deployment. The merit of this scheme is that it provides excellent resilience against node capture. It also provides node-to-node authentication. The demerit of this scheme is that it does not have good scalability and has additional overhead needed for each node to maintain and store distinct pair-wise keys. It is difficult to add new sensor nodes in the network because the key of the new node has to be store in each of the existing node. The random key distribution, using a unique pool, helps to establish high network connectivity. It is not resistant against node capture because the attacker will reveal *k* keys, and each one of these keys may be used to compromise one or more channels in the WSN.

2.3.3. Pure Probabilistic Key Pre-Distribution Schemes: The master key based schemes and pair-wise key pre-distribution scheme are trivial solutions. Both are having more disadvantages than advantages. In both the cases if memory requirement and key connectivity is considered then resilience would compromise or vice versa. Thus to overcome such disadvantages, there is one more solution which ensures some probability that any two sensor nodes can communicate using a pairwise key. The scheme does not, however, ensure that two nodes always are able to compute a pairwise key to use for secure communication.

Random Key Pre-Distribution Scheme [13]: A random key pre-distribution scheme was proposed for distributed sensor a network, which is dynamic in nature, in the sense that they allow addition and deletion of sensor nodes in the network after deployment. The scheme addresses the bootstrapping problem. So, that the network grows or replacing failing and unreliable nodes. They have solved the problem of knowing the exact set of neighbors by knowing the set of possible or likely neighbors for each node. The reason is that the scheme guarantees only any two neighboring nodes can find a common secret key with a certain probability p, instead of 100% certainty. Since, the approach of the scheme is the probabilistic key sharing among the nodes of a random graph and uses simple protocols for shared-key discovery and path-key establishment, and for key revocation, re-keying, and incremental addition of nodes. The key-distribution process consists of three phases: (i) key pre-distribution, the key pre-distribution phase further consists of five off-line steps (a) generation of a large pool of p keys, and their key identifiers; (b) random selection of k keys out of P keys to establish a key-ring; (c) loading the key-ring to each sensor node; (d) saving the key identifiers of a key-ring and associated sensor identifier on a trusted controller node; (e) for each node, loading the i-th controller node with the key shared with that node.; (ii) shared-key discovery where every node discovers its neighbors in wireless communication range with which it shares keys.; (iii) path-key establishment phase gives a path-key to selected pairs of

sensor nodes in wireless communication range that do not share a key but are connected by two or more links at the end of the shared-key discovery phase. A simulation has been done based on various parameters; like efficiency, scalability and diameter of the resulting secure network; on different distributed sensor network sizes.

Q-Composite Random Key Pre-distribution Scheme [15]: In q-composite random key pre-distribution scheme q-common keys from the key pool instead of one common key in the basic scheme are employed in each of the sensor node. This will increase the key overlap required for key-setup, thus increasing the resilience of the network against node capture. Hence, the scheme strengthened the security under small-scale attack. But vulnerability increases when the scheme has to face large-scale physical attack.

Multipath Key Reinforcement Scheme [15]: The scheme coordinates the key updates over multiple independent paths. They assumed that after key-setup multiple secure paths are formed because of q common keys shared by the nodes. That means once a secure link is formed, the communication key between nodes must be updated when one is compromised. This should be coordinated using multiple independent paths for greater security but not done via the already established link, as an adversary might decrypt the communication to obtain the new key. The scheme increases the security of key setup such that an attacker has to compromise many more nodes to achieve a high probability of compromising any given communication. The scheme offers better security than the Basic Scheme [13] or the Q-Composite [15]. But it creates communication overhead that can be reason to deplete battery life of the node and gives the chance for an adversary to launch DOS attacks. The scheme improves the resilience to node capture.

Closest Pairwise Keys Pre-Distribution Scheme [3, 16]: A scheme closest pairwise keys pre-distribution scheme was proposed by taking advantage of sensors' expected locations. In this scheme, to have each sensor share pairwise keys with c other sensor nodes whose expected locations are closest to the expected location of the sensor, where c is a system parameter determined by the memory constraint. They started with a basic version, which is a combination of the random pairwise keys scheme [15] and the location information. Then they presented an extended version to further to reduce the storage overhead and facilitate dynamic deployment of new sensors. They considered the following assumptions: (1) setup server is responsible for key pre-distribution. (2) setup server is aware of the network-wide signal range and deployment error, and the expected location of each sensor before deployment. (3) each sensor has a unique, integer-valued ID. The basic idea of the scheme is to pre-distribute pairwise keys between pairs of sensors. So that two sensors have a pre-distributed pairwise key if they have a high probability to appear in each other's signal range. The idea is difficult to implement. The reason is that it is non-trivial to get the probability that two sensors are neighbors. So, they pre-distributed pairwise keys between pairs of sensors whose expected locations are close to each other more physically located in each other's signal range. Further, they gave the extended version of this scheme based on the two limitations of the previous scheme. (1) if the sensors are not evenly distributed in the target field then it is possible for a sensor to have a large number of neighbor sensors that are not among the closest c sensors. Consequently, a sensor has to store a lot of pairwise keys. (2) to add a new sensor after deploying the sensor network, the setup server has to inform a number of existing sensors in the network about the addition of the new sensor, which may introduces a lot of communication overhead. So in the extended version they proposed an alternative way to pre-distribute the secret information. The technique is based on a pseudo random function (PRF) [1] and a master key shared between each sensor and the setup server. The extended scheme has additional overhead by requiring master sensors to remember the IDs of

their revoked slaves. The advantage of this scheme is that the compromise of sensor nodes does not lead to the compromise of direct keys shared between non-compromised sensor nodes.

Random Key Pre-Distribution Scheme Using Node Deployment Knowledge [23]: A random key pre-distribution scheme was proposed which is using node deployment knowledge. The authors have argued for key pre-distribution, information of the closest neighbors is important. Thus, the most important knowledge for pre-distribution is the knowledge of the nodes that are likely to be neighbors of each sensor nodes. Deployment knowledge can be exhibited using probability density functions (pdfs). When the pdf is uniform, no information can be gained on where a node is more likely to reside. All the existing pre-distribution key schemes consider uniform distribution. They considered non-uniform probability density functions (pdfs). They have shown that knowledge would improve random key pre-distribution scheme [13]. Thus, the scheme minimizes the number of keys, increases resilience to the node capture and reduces network overhead; it also increases overall connectivity of the network graph. This key management scheme takes advantage of knowing the position of a sensor within a sequence prior to its deployment. The scheme considers non-uniform probability density functions, which means that they assume the positions of sensor nodes to be at certain areas. But, the scheme includes complexity.

Key Pre-Distribution Using Post-deployment Knowledge [3]: A key pre-distribution scheme was proposed which is using a concept of post deployment knowledge of sensor nodes to improve the pairwise key pre-distribution in static sensor networks. The idea behind the concept is to assign each sensor node an excessive amount of pre-distributed keys by using the memory for sensing applications. Then prioritize the pre-distributed keys based on post-deployment knowledge, and discard low priority keys to thwart node compromise attacks and return memory to the applications. They called this process as key prioritization. They proposed to remove the keying materials that are less likely to be used (based on the post-deployment knowledge). Since the compromising of a sensor node reveals more secrets in the network. They further assumed that an attacker couldn't recover the removed keys at sensor nodes even if these nodes are compromised later. So, they presented an approach of key prioritization in static sensor networks, and then showed the improvement of the random subset assignment scheme [17] using the proposed approach. Results showed security is improved significantly as the additional memory for polynomial shares at the pre-distribution stage increases. Second improvement is that the period of time after deployment and before key prioritization becomes shortened which is the most vulnerable period. Storage overhead is also reduced; now, two sensor nodes only need to exchange their polynomial IDs.

2.3.4. Polynomial-Based Key Pre-Distribution Schemes: Polynomial-based key pre-distribution scheme is based on pairwise keys pre-distribution schemes. Thus these schemes overcome some of the probabilistic pre-distribution schemes' disadvantages. These are: (1) Any two sensors can definitely establish a pairwise key when there are no compromised sensors; (2) Even with some nodes compromised, the others in the network can still establish pairwise keys; (3) A node can find the common keys to determine whether or not it can establish a pairwise key and thereby help reduce communication overhead.

Polynomial Based Pairwise Key Pre-Distribution [17]: A key pre-distribution scheme was proposed which is using the concept of polynomials. The scheme uses the concept of the protocol [11]. The protocol in [11] was developed for group key pre-distribution. But the author needs to establish pairwise keys, for simplicity, they discuss the special case of pairwise key establishment in the context of sensor networks. To pre-distribute pairwise keys,

the (key) setup server randomly generates a bivariate *t*-degree polynomial $f(x,y) = \sum_{i,j=0}^{t} a_{ij} x^i y^j$ over a finite field F_q where q is a prime number that is large enough to accommodate a cryptographic key, such that it has the property of $f(x, y) = f(y, x)$. They assumed that each sensor has a unique ID. For each sensor i, the setup server computes a *polynomial share* of $f(x, y)$, that is $f(i, y)$. For any two sensor nodes i and j, node i can compute the common key $f(i, j)$ by evaluating $f(i, y)$ at point j, and node j can compute the same key $f(j, i) = f(i, j)$ by evaluating $f(j, y)$ at point i. The advantage of this scheme is each sensor node i needs to store a t-degree polynomial $f(i, x)$, which occupies $(t+1)\log q$ storage space. For the establishment of the pairwise key both sensor nodes need to evaluate the polynomial at the ID of the other sensor node.

Polynomial Pool-Based Pairwise Key Pre-Distribution [17]: A general framework for key pre-distribution was developed which is a combination of polynomial based key pre-distribution and key pool [13, 15]. The polynomial based key pre-distribution scheme has a disadvantage that it can only tolerate no more than t compromised nodes, where the value of t is limited by the memory available in sensor nodes. In this framework a pool of randomly generated bivariate polynomials is used to establish pairwise keys between sensors. The polynomial pool has two special cases. (1) When the polynomial pool has only one polynomial, the general framework degenerates into the polynomial-based key pre-distribution. (2) When all the polynomials are 0-degree ones, the polynomial pool degenerates into a key pool as in Basic Scheme [13] or the Q-Composite scheme [15]. Pairwise key establishment in this framework is performed in three phases: setup, direct key establishment, and path key establishment. The drawback is that such methods may introduce substantial communication overhead.

Random Subset Assignment Key Pre-Distribution Scheme [4, 17]: An efficient instantiation of the general framework polynomial pool-based pairwise key pre-distribution was developed. In this scheme a random strategy is used for subset assignment during the setup phase. A random subset of polynomials in F is selected by the setup server and a polynomial share is assigned to the sensor node. The scheme is an extension of the basic probabilistic scheme [13]. As compared to basic scheme this scheme uses unique key between each pair of sensors.

A set F of s-bivariate t-degree polynomials over F is generated by setup server and a subset of s' polynomials of F is selected and polynomial share is assigned to each node. The probability that two sensors establish a pairwise key directly can be estimated by

$$p = 1 - \prod_{i=0}^{s'-1} \frac{s - s' - i}{s - i}$$

A comparison is being made with basic probabilistic scheme, q-composite scheme and random pairwise key scheme and concluded that number of compromised nodes in this scheme is less. Storage overhead is also low. Moreover, sensors can be added without having to contact already deployed nodes.

Grid-Based Key Pre-Distribution [17]: A scheme was proposed which is based on the components of the general framework proposed by them. The scheme guarantees that any two sensors can establish a pairwise key when there is no compromised sensor, provided that the sensors can communicate with each other. The scheme is resilient against node capture. There is a high probability to establish a pairwise key between non-compromised sensors. Moreover

in the scheme, sensor can able to determine the establishment of a pairwise key with another node. It also knows how to compute the pairwise key. This would reduce the communication overhead during the discovery of shared keys.

Location-Based Pair-Wise Keys Scheme Using Bivariate Polynomials [16]: A Location-Based Pair-Wise Keys Scheme Using Bivariate Polynomials was proposed which is based on polynomial-based key pre-distribution technique and closest pairwise keys scheme. In this scheme, the target field partitioned into small areas called *cells*, each of which is associated with a unique random bivariate polynomial. Then a set of polynomial shares that belong to the cells closest to the one that the sensor is expected to locate in is distributed to each sensor instead of assigning each sensor the pairwise keys for the closest sensors. The target field is assumed to be a rectangular area that can be partitioned into equal-sized squares.

Closest Polynomials Scheme [3]: A closest polynomials scheme was presented which is a combination of the expected locations of sensor nodes with the random subset assignment scheme in [17] to overcome certain limitations. The limitations are the constraints on the storage capacity, node density, signal range and deployment pdf (probability density functions), the probability of establishing direct keys are fixed. The idea behind the scheme is to select polynomials for each sensor node based on its expected location instead of randomly selecting polynomials for each sensor node as in the original random subset assignment scheme. In this scheme, the target field partitioned into small areas called *cells*, each of which is associated with a unique random bivariate polynomial. Then a set of polynomial shares that belong to the cells closest to the one that the sensor is expected to locate in is distributed to each sensor. The target field is assumed to be a rectangular area that can be partitioned into equal-sized squares. The result shows that for getting higher security, sensor deployment must be more precise.

Hypercube-Based Key Pre-Distribution Scheme [4]: The scheme is a generalization of grid-based key pre-distribution scheme [17]. The scheme has some attractive properties. (1) It guarantees that any two sensor nodes can establish a pairwise key when there are no compromised sensor nodes, assuming that the nodes can communicate with each other. (2) The scheme is resilient to node compromises even if some nodes are compromised; there is still a high probability to re-establish a pairwise key between two non-compromised nodes. (3) A sensor node can directly determine whether it can establish a direct key with another node, and if it can, which polynomial should be used. Thus, there is no communication overhead during polynomial share discovery. For a N number of sensor nodes in the network, the hypercube-based scheme constructs an n-dimensional hypercube with m^{n-1} bivariate polynomials arranged for each dimension $j, \left\{ f^{j}_{\langle i_1,\ldots,i_{n-1}\rangle}(x,y) \right\} 0 \le i_1,\ldots,i_{n-1} < m$, where $m = \left\lceil \sqrt[n]{N} \right\rceil$. Each sensor node stores n polynomial shares and each polynomial is shared by about m different nodes. Each node can establish direct keys with $n(m-1)$ other nodes. Thus, the scheme is having less memory requirement, computing a pairwise key in our schemes can be faster, so less computation cost.

Random Perturbation-Based (RPB) Scheme [27]: A random perturbation-based (RPB) scheme was proposed for pairwise key establishment in sensor networks. The scheme is based on polynomials to generate pairwise keys. The polynomials are defined over a finite field denoted as F_q, where q is a prime number. The perturbation polynomial is defined as: Given a finite field F_q, a positive integer r ($2^r < q$), and a set of node IDs $S \left(S \subseteq \{0,\ldots,q-1\}\right)$, a polynomial set Φ is a set of perturbation polynomials regarding r and S if any polynomial

$f(x) \in F$ has the following limited infection property: $\forall u \in S, f(u) \in \{0, \ldots, 2^r - 1\}$. Whereas the value of a perturbation polynomial will not be greater than $2^r - 1$; i.e., it has at most *r* bits. The RPB scheme does not give each node the original share but the perturbed share, which is the sum of the original share and a perturbation polynomial with the limited infection property. The motivation behind the adding of perturbation with limited infection is: (1) it makes harder to break the symmetric polynomials. The reason is that the adversary cannot obtain the original shares of polynomial $f(x, y)$, (2) Two nodes can still establish a key. The addition of perturbation polynomials changes the values of the original match key $(f(u, v))$at both sides. The scheme is having low computation and communication overhead. The scheme is highly efficient and low storage requirement.

2.3.5. Matrix-based key pre-distribution schemes: In matrix-based key pre-distribution schemes, all possible link keys in a network of size *n* can be represented as an $n \times n$ key matrix. Small amount of information is stored to each sensor node, so that every pair of nodes can calculate corresponding field of the matrix, and uses it as the link key. In this scheme [10], a symmetric matrix K_{nxn} stores all pairwise keys of a group of *n* nodes, where each element k_{ij} is the key of node *i* for securing the link with node *j*. $k = (DG)^T G$, Where $D_{(l+1)\times(l+1)}$, is symmetric and $G_{(l+1)\times n}$ is called public matrix, and $(DG)^T$ is called secret matrix and must be kept secret to all nodes.

Grid-Group Deployment Scheme [22]: Location-aware key management scheme was proposed which is known as grid-group deployment scheme. Sensor nodes are uniformly deployed in a large area instead of randomly distributing keys from a large key pool to each sensor. Secret keys are systematically distributed to each sensor from a structured key pool. The performance analysis shows that scheme requires less number of keys preinstalled for each sensor and is resilient to selective node capture attack and node fabrication attack.

Robust Group-Based Key Management Scheme [25]: A scheme was proposed which is known as group-based key management scheme using sensor deployment knowledge based on Blom' scheme [10]. According to the scheme the sensor field is partitioned into hexagonal grids. On the basis of the arrangement sensor nodes are deployed. The scheme achieved a higher degree of connectivity with the help of deployment knowledge as compared to the basic scheme. The scheme reduced the number of potential neighbors of each node, thus, lower the memory requirement of the nodes. To generate pairwise keys for nodes, the scheme distributes secret information instead of secret keys in sensor nodes. For the efficiently generation of pair-wise keys for neighboring nodes, they assigned each group a distinct secret matrix and made neighboring groups share some other secret matrices. Simulation results showed that scheme is having higher degree of connectivity in sensor networks with a lower memory requirement, good resilience against node capture attacks as compared to the existing schemes.

Multiple-Space Key Pre-distribution Scheme [5]: A pairwise key pre-distribution scheme was proposed which is based on Blom's key pre-distribution scheme [10] and combines the random key pre-distribution method [13] with it; which offers improved network resilience known as Multiple-Space Key Pre-distribution Scheme. The idea behind this is to convert the complete graph to a connected graph, so that each sensor node needs to carry less key information. For this they use the concepts from graph theory and draw an edge between two nodes if and only if they can find a secret key between themselves. They defined a key space as a tuple *(D, G)*, where matrices *D* and *G* are as defined in Blom's

scheme. Scheme has two phases: (1) Key Pre-distribution Phase, (2) Key Agreement Phase. Results showed that scheme is scalable and more resilient.

ConstrAined Random Perturbation based pairwise keY establishment (CARPY) scheme [28]: ConstrAined Random Perturbation based pairwise key establishment scheme and its variant CARPY+ was proposed for Wireless Sensor Networks. The proposed CARPY+ scheme meets the requirements: 1) resilience to the adversary's intervention, 2) directed and guaranteed key establishment, 3) resilience to network configurations, 4) efficiency, and 5) resilience to dynamic node deployment. The proposed schemes generalize and improve Blom's concept. To enhance the security in Blom's concept [10] the direct relation between *D* and *A* has been split by adding certain random perturbation on *A* to distort Blom's key. If improper random perturbation is applied, either additional computation and communication are needed to extract the common bits of distorted Blom's key between two sensor nodes, or the common key cannot be found anymore. So, they proposed *constrained random perturbation* (CRP). In the CARPY scheme, there are two steps: (1) the off-line step- is performed, before deployment of sensor nodes, to determine the desired key length, select appropriate parameters, and pre-install the keying materials into the sensor nodes. (2) the on-line step- is performed for each pair of sensor nodes required to find the pairwise key in common after sensor nodes are deployed. The second variation is (*CARPY+*) *Communication-Free CARPY Scheme.* In the CARPY scheme, two sensor nodes communicate with each other only for exchanging the respective column of *G*, which can be known by the adversary. If each column of *G* can be generated by each sensor node itself, then communications will no longer be necessary. The only requirement for *G* is that any λ + 1 columns of *G* should be linearly independent. Thus, the Vandermonde matrix is most suitable because, if ϕ is the primitive element of F_q, then any λ+1 columns of Vandermonde matrix, which is generated by only one element j , are linearly independent. Note that such Vandermonde matrix is of the form that the *i*-th column is generated by $\left[1 \; j^{i} \; \left(j^{i}\right)^{2} \; \ldots \; \left(j^{i}\right)^{\lambda}\right]$, where λ is a security parameter independent of *N*. Therefore, communication overhead can be eliminated if the matrix *G* of CARPY is selected as a Vandermonde matrix. For convenience, the CARPY scheme with *G* being a Vandermonde matrix is called CARPY+.

2.3.6. Tree-based key pre-distribution schemes*:* In tree-based key pre-distribution schemes, senor nodes are arranged in a tree in which each sensor node communicates with its parent node. So the key establishment has done between neighboring nodes along the aggregation tree. The new node receives two tickets that can be verified by two existing nodes randomly selected by the network administrator, before joined in a network. After the deployment of a new node into the network, it generates a pairwise key for its parent node. For securely transmitting the key to the parent, the new node splits the key into two parts and sends them with its tickets to the nodes selected by the administrator, which authenticates the new node and forwards key materials to the parent of the new node. The merit of a tree-based key distribution is the significantly reduction of the memory cost.

ID-Based On-Way Function Scheme [9]: A basic ID-based one way function scheme (IOS) was proposed which is having a tree-based key pre-distribution. Basically, authors focused over the problem of resiliency against coalition attack and show that these schemes are better than the existing ones. In the scheme, a public one way hash function is used in order to reduce the number of keys stored in the node. A unique ID is assigned to each sensor node and this ID is used to compute secret keys. A key pool $K = \left\{k_v, v \in G\right\}$ is constructed after determining a network graph G. Now, decompose the edge of graph G into star like sub-

graphs. If a node is contained in a star like sub-graph centred at v, node u receives a secret key k_u and hashed key $h(k_v \| ID(u))$. Now, u and v can establish a shared key $h(k_v \| ID(u))$. Since basic IOS is not suitable for large size network, multiple IOS is used by using multiple copies of a single IOS. Basic IOSs are not suitable for a network of large size since they can accommodate only $O(k)$ sensor nodes for the node storage of k keys. They enhance their scheme to multiple ID-based one way function scheme. In the scheme a multiple copies of a single basic IOS is used to increase the network size relative to available memory. This will weakened the resiliency of multiple IOSs. A (m,r,l,m)- strongly regular graph *G*, which is decomposed by star-like sub-graphs is determined to deploy $n = ml$ sensor nodes.

Deterministic Multiple Space Blom's Scheme [10]: A deterministic multiple space Blom's scheme (DMBS) was proposed to improve the resiliency of the multiple IOSs. In order to achieve good resiliency they weakened the connectivity of the network graph. They considered the complete bipartite graph $K_{m1,m2}$ instead of a complete graph. In the scheme l copies of a (m,r,l,m)- strongly regular graph G are considered to accommodate $n = ml$ nodes. They regard each vertex of G as a class of l nodes. Every sensor node u_i receives its public column vector x_{ui} from a Vandermonde matrix M and every edge e of G is associated with a random $(t + 1) \times (t + 1)$ matrix D_e, not necessarily symmetric.

2.3.7. Hierarchical Key Management Scheme: A tree of keys is built for the hierarchical network, where the keys at a certain level are distributed to the corresponding class of nodes. The keys at higher levels can be used to derive the keys at lower levels, but not vice versa. The intention of the hierarchical network is to facilitate data collection and fusion and query propagation in hostile environments.

Localized Encryption and Authentication Protocol (LEAP) [20]: A protocol was proposed known as Localized Encryption and Authentication Protocol. The protocol was designed to support in-network processing. The protocol restricts the security impact of a node compromise to the immediate network neighbourhood of the compromised node. LEAP supports the establishment of four types of keys for each sensor node – an individual key shared with the base station, a pairwise key shared with another sensor node, a cluster key shared with multiple neighboring nodes, and a group key that is shared by all the nodes in the network. The protocol provides the basic security services such as confidentiality and authentication. For the establishment of the individual node key, every node shares a single shared key with the base station. This key is generated and pre-loaded into each node prior to its deployment. In this, the computational overhead is negligible due to the computational efficiency of pseudo random functions. For the establishment of the pairwise shared key; it can be done by pre-loading the sensor nodes with the corresponding pairwise keys. In any sensor network applications, a sensor node always communicates with its immediate neighbors. Therefore, this type of pairwise keys is most commonly used. If a sensor network application has special requirements of establishing pairwise keys for two nodes that are multiple hops away, the multiple-path scheme [15] or a probabilistic key sharing scheme [16, 19] may be employed. Establishment of pairwise keys is done in a four step process a: Key Pre-distribution, Neighbor Discovery, Pairwise Key Establishment and Key Erasure. The pairwise key establishment scheme incurs computational overhead. Since single key preloaded in the node, thus it requires less space for storing keys. Hence the computational, communication, and storage overhead of the scheme for pairwise key establishment is very small. For the establishment of the cluster key, this phase follows the pairwise key establishment phase. The node first generates a random key then encrypts this key with the

pairwise key shared with each neighbor, and then transmits the encrypted key to each neighbor. Then the neighbor decrypts the key stores it in a table, and then sends back its own cluster key to node. When one of the neighbors is revoked, node generates a new cluster key and transmits it to all the remaining neighbors in the same way. A group key is a key shared by all the nodes in the network. To bootstrap a group key for a sensor network is to pre-load every node with the group key. It is important to securely update the group key when a compromised node is detected. The group key must be changed and distributed to all the remaining nodes in a secure, reliable, and timely fashion. This is referred to as group rekeying. LEAP also includes an efficient protocol for local broadcast authentication based on the use of one-way key chains. An analysis has been done on the security of LEAP under various attack models and result shows that LEAP is very effective in defending against many sophisticated attacks such as HELLO Flood attack, Sybil attack, and Wormhole attack.

A Time-Based Deployment Model [26]: A Time-Based Deployment Model was proposed that would localize the impact of key compromise within the time intervals. The time for sensor nodes to establish pair-wise keys each other, T_{est}, may take longer than the time interval for an adversary to compromise a node, T_{min}. If an initial key K_I is disclosed within T_{min} time, then the whole sensor network is threatened by an attacker. Even though an initial key K_I is disclosed by an attacker, the portion of network compromised must be minimized. For that reason they split the time domain to disperse the damage resulting from the disclosure of an initial key K_I. Selection of K_I with probabilistic time intervals is as follows: Key Setup Phase, Initial Key Establishment Phase, Node Addition Phase. They have concluded that the multiple keying mechanism of LEAP satisfies the multiple usages of sensor networks, but they have found that LEAP actually missed a possible disclosure of an initial key K_I. While the initial deployment phase is assumed secure and the security mainly relies on the initialization key, so the key is erased from sensor nodes after the initialization phase. However, the same key should be used again for node addition after that phase while the new node can be captured before removing the initialization key.

2.3.8. Combinatorial Design-based key Pre-distribution Scheme [8, 21]: A key deterministic and hybrid pre-distribution scheme was proposed based on Combinatorial Design theory that decides how many and which keys to assign to each key-chain before the sensor network deployment. Combinatorial design theory is interested in arranging elements of a finite set into subsets to satisfy certain properties. A *BIBD* is an arrangement of *v* distinct objects into *b* blocks such that each block contains exactly *k* distinct objects, each object occurs in exactly *r* different blocks, and every pair of distinct objects occurs together in exactly λ blocks. The design can be expressed as (v,k,l) or equivalently (v,b,r,k,l), where: $l(v-1)=r(k-1)$ and $bk=vr$. A *BIBD* is called Symmetric BIBD or Symmetric Design when b = v and therefore $r = k$. A Finite Generalized Quadrangle *GQ (s, t)* is an incidence structure $S = (P,B,I)$ where *P* and *B* are disjoint and nonempty sets of points and lines respectively, and *I* is a symmetric point-line incidence relation satisfying the following axioms: (1) Each point is incident with $t+1$ lines $(t \geq 1)$ and two distinct points are incident with at most one line, (2) Each line is incident with $s+1$ points $(s \geq 1)$ and two distinct lines are incident with at most one point, (3) If *x* is a point and *L* is a line not incident (I) with x, then there is a unique pair $(y, M) \in PXB$ for which $x\ I\ M\ I\ y\ I\ L$. A Symmetric Designs and Generalized Quadrangles are mapped to generate key-chains to obtain efficient key distribution schemes for the sensors in a sensor network. The main drawback of the combinatorial approach comes from the difficulty of its construction. To overcome the drawback, they proposed a Hybrid Design

which combines a deterministic core with a probabilistic extension. They considered two Hybrid Designs: Hybrid Symmetric and Hybrid GQ Designs.

2.3.9. EBS-Based Key Pre-Distribution Schemes [2]: The scheme is a combinatorial optimization methodology for key management of group communication setups. The EBS scheme exploits the trade-off between the number of administrative keys *k* and the number of rekeying messages *m*. A set of $\left(k+m\right)$ administrative keys is used to support a set of N nodes, where each node is assigned a distinct combination of *k* keys. A node can be simply admitted to the group by assigning one of the unused set of *k* keys out of the total of $C\left(k+m,k\right)$ i.e. $\left(k+m\right)!/\left(k!m!\right)$ distinct combinations. Eviction of a compromised node can be performed by broadcasting replacement of the *k* keys that the evicted node knows using the *m* keys that the node does not know. The EBS approach proves to be very scalable for large networks and enables great flexibility in network management by controlling the values of *k* and *m*. Large *k* increases the storage requirements at the node, while large *m* increases communication overhead for key management. An EBS is defined as a collection Γ of subsets of the set of members. Each subset corresponds to a key and the elements of a subset $A \hat{\mathrm{I}}\ \mathrm{G}$ are the nodes that have that key.

Scalable, Hierarchical, Efficient, Location-aware, and Light-weight (SHELL) Scheme [7]: A distributed key management scheme was proposed based on Exclusion Basis Systems (EBS); a combinatorial formulation of the group key management problem Networks that performs location based key assignment to minimize the number of keys revealed by capturing collocated nodes. The scheme is known as SHELL - Scalable, Hierarchical, Efficient, Location-aware, and Light-weight. The scheme supports rekeying thus enhances network security and survivability against node capture. SHELL uses the EBS framework to perform rekeying within each cluster. Cluster gateways keep track of the key assignment but not the actual keys. SHELL is collusion-resistant. The scheme uses post-deployment location information in key assignment; collocated nodes share more keys than nodes that are not collocated. The scheme distributes key management functionality among multiple nodes and minimizes the memory and energy consumption through trading off the number of keys and rekeying messages. The scheme employs a novel key assignment scheme that reduces the potential of collusion among compromised sensor nodes by factoring the geographic location of nodes in key assignment.

LOcalized Combinatorial Keying (LOCK) scheme [6]: A dynamic key management scheme was presented called LOCK (LOcalized Combinatorial Keying) for clustered sensor network. LOCK performs localized rekeying to minimize overhead. The physical network model is a three-tier wireless sensor network with the base station (BS) at the top, followed by cluster leader nodes (CLs), then regular sensor nodes. In LOCK, no pre-deployment information is assumed about the expected locations of the nodes. LOCK uses two layers of EBS administrative keys. The upper layer (level 1) is EBS_b that enables the base station to manage the cluster leaders as a group. LOCK is an improved scheme for SHELL [7]. LOCK uses the key polynomials to improve network resilience to collusion instead of location-based key assignment as in SHELL.

3. Conclusion

Key management for wireless sensor networks is one of the main concerns in terms of providing security. The study of key management in wireless sensor networks still has abundant research opportunities in the future. As for now, key management systems are a trade-off of performance and security to low overhead in memory usage and message transmissions. Key management systems sole purpose is to supply secure communication in wireless sensor networks without producing much overhead. This paper provides an overview of various techniques proposed by different researchers. Also, we provide taxonomy of symmetric key management schemes for wireless sensor networks and their advantages and disadvantages. We classify symmetric key management schemes broadly into three categories, these are: base-station participation scheme, trusted third node based scheme and pre-distribution schemes. Based on the key distribution, key discovery and key establishment in the pre-distribution schemes, we classify these schemes into nine categories, these are: master key based pre-distribution scheme, pairwise key pre-distribution schemes, pure probabilistic key pre-distribution schemes, polynomial based key pre-distribution schemes, matrix based key pre-distribution schemes, tree based key pre-distribution schemes, hierarchical key management scheme, combinatorial design based key pre-distribution schemes, EBS based key pre-distribution schemes.

Table 1. Merits and Demerits of Symmetric Key Management Schemes

Scheme	Merits	Demerits
SPINS [12]	- Good resilient to node capture - Possible to revoke key pairs	- Not scalable - Base station becomes the target of attacks
LKHW [18]	- Minimal storage	- Base station becomes the target of attacks
PIKE [24]	- low communication overheads as compared to random key pre-distribution scheme - key establishment is not probabilistic	- Intermediary node may be the target of attack
BROSK [14]	- minimal storage - less computations - no need to perform key discovery or key exchange	- very low resilience - Single node causes the compromise of the entire network through the shared key
Lightweight Key Management System [29]	- Low memory and computation cost - Low key setup overhead - Self-organizing	- Low resilience because an adversary only needs to compromise the *authentication key* and a *key-generation key* of that generation to compromise all the links of nodes in that generation - Adversary may log the messages flowing in the network to process later when the required credentials are compromised completely
Random Pair-Wise Key [15]	- node-to-node authentication - High network connectivity	- limited scalability - less resilient - additional overhead for each node to maintain and store distinct pair-wise keys
Random Key Pre-Distribution [13]	- good network connectivity - provides revocation, re-keying	- not resistant against node capture
Q-Composite Random Key Pre-distribution [15]	- improved resilience than random key pre-distribution scheme	- less probability of key sharing than random key pre-distribution scheme

Multipath Key Reinforcement [15]	- Better security than the random key pre-distribution or the Q-Composite scheme - improved resilience	- creates communication overhead - depleted node battery life
Closest Pair-wise Keys Pre-Distribution [3, 16]	- knows sensors' expected locations - compromise of sensor nodes does not lead to the compromise of direct keys shared between non-compromised sensor nodes - facilitate dynamic deployment of new sensors - reduced storage overhead	- additional overhead by requiring master sensors to remember the IDs of their revoked slaves - sensors are not evenly distributed in the target field - communication overhead-to add a new sensor after deploying the sensor network
Random Key Pre-Distribution Scheme Using Node Deployment Knowledge [23]	- good storage complexity - good resilience - reduced overhead - good connectivity of the network graph	- complex
Key Pre-Distribution Using Post-deployment Knowledge [3]	- improved pair-wise key pre-distribution in static sensor networks - good security - reduced storage overhead; two sensor nodes only need to exchange their polynomial IDs	- complex; keep track of locations of nodes before and after deployment
Polynomial Based Pair-wise Key Pre-Distribution [17]	- need to evaluate the polynomial at the ID of the other sensor node - good resilience	- poor key connectivity - storage overhead - tolerate no more than t compromised nodes
Polynomial Pool-Based Pair-wise Key Pre-Distribution [17]	- good resilience - good scalability - low communication overhead - low computation overhead - total connected graph	- poor key connectivity - Storage overhead
Random Subset Assignment Key Pre-Distribution Scheme [4, 17]	- good resilience - good scalability	- less secure - if an adversary knows the distribution of polynomial shares, specific nodes can be targeted for attack to compromise communications
Grid-Based Key Pre-Distribution[17]	- low communication overhead - low computation overhead - total connected graph	- storage overhead; as each node share the polynomial keys and the ID's
Location-Based Pair-Wise Keys Scheme Using Bivariate Polynomials [16]	- increase the direct connectivity ratio of neighboring nodes - allow neighboring nodes to communicate with each other with higher probability	- anchor nodes are randomly dispersed in the environment causes denial of service attack
Closest Polynomials Scheme [3]	- compromise of sensors does not lead to the compromise of direct keys shared between non-compromised sensors - tolerate a large fraction of compromised nodes - probability is independent from the total number of polynomials in the pool	- cell side length is large
Hypercube-Based Key Pre-Distribution Scheme [4]	- much more efficient than grid based key pre-distribution schemes	- more complex

Random Perturbation-Based (RPB) [27]	- highly secure - less computations - less storage requirement	- more energy consumption - less efficient
Grid-Group Deployment [22]	- uniformly deploys sensors in a large area rather than random key pre-distribution schemes - less number of keys preinstalled	- based on assumption of random capture is too weak
Robust Group-Based Key Management [25]	- higher degree of connectivity - lower memory requirement - stronger resilience - scheme distributes secret information instead of secret keys	- center of a grid is a deployment point; target of attack
Multiple-Space Key Pre-distribution [5]	- requires much lower memory usage - scheme is scalable and flexible, and nodes do not need to be deployed at the same time	- not optimal resilience
Constrained Random Perturbation Based Pair-wise Key Establishment (CARPY) [28]	- great resilience - first non-interactive key establishment scheme	- energy consumption is more
ID-Based On-Way Function [9]	- Multiple copies of single basic IOS would increase the network size -	- not suitable for large size network - accommodate only $O(k)$ sensor nodes for the node storage of k keys - Multiple copies of single basic IOS would decrease the resilience
Deterministic Multiple Space Blom's [10]	- l copies improves the resiliency against multiple IOS	- -
Localized Encryption and Authentication Protocol (LEAP) [20]	- The use of different keys to secure different types of communication channels in the WSN may increase the overall efficiency of the communications; it leads to a very low resistance degree.	- The entire network will suffer a severe loss if an initial key is exposed to an attacker during key setup. - Hence, early key establishment must be completed quickly in order to strengthen the security of LEAP.
A Time-Based Deployment Model [26]	- uses probabilistic time intervals - localize the impact of K_I disclosure within the time intervals	- the assumption of security in the initial deployment phase is not viable in many cases since the initial deployment of dense networks may not take short as LEAP expected.
Combinatorial Design-Based Key Pre-Distribution [8, 21]	- it increases the probability of a pair of sensor nodes to share a key - decreases the key-path length while providing scalability with hybrid approaches	- hybrid design sacrifices key sharing probability
Exclusion-Based System (EBS) [2]	- Good network survivability - Good scalability - secure and efficient	- If small number of nodes in the network is compromised, information about the entire network could be uncovered
SHELL [7]	- If the gateway is compromised, the command node is notified. - If sensor node is compromised, it is given that the gateway node can detect it.	- Nodes need to be in direct transmission range of each other to collude. - structure and operation are highly complex, involving heterogeneous node operations and multiple (at least seven) types of keys.
LOCK [6]	- LOCK uses key polynomials to improve network resilience to collusion instead of location-based key assignment as in SHELL - the capture of any node in LOCK (including cluster leaders) does not affect the normal operation of other clusters.	- Includes rekeying overhead

References

[1] O. Goldreich, S. Goldwasser and S. Micali, "How to Construct Random Functions", J of the ACM, vol. 4, no. 33, **(1986)**, pp. 792–807.
[2] M. Eltoweissy, H. Heydari, L. Morales and H. Sudborough, "Combinatorial Optimization of Group Key Management", J Network and System Management, vol. 1, no. 12, **(2004)**.
[3] D. Liu and P. Ning, "Improving Key Pre-Distribution with Deployment Knowledge In Static Sensor Networks", ACM Transactions on Sensor Networks, vol. 1, no. 2, **(2005)**, pp. 204–39.
[4] D. Liu, P. Ning and R. Li, "Establishing Pairwise Keys in Distributed Sensor Networks", ACM Transactions on Information and System Security, vol. 1, no. 8, **(2005)**, pp. 41–77.
[5] W. Du, J. Deng, Y. S. Han, P. Varshney, J. Katz and A. Khalili, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks", ACM Transactions on Information and System Security (Tissec), vol. 2, no. 8, **(2005)**, pp. 228–58.
[6] M. Eltoweissy, M. Moharrum and R. Mukkamala, "Dynamic Key Management in Sensor Networks", IEEE Communications Magazine, **(2006).**
[7] M. F. Younis, K. Ghumman and M. Eltoweissy, "Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Network", IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 8, **(2006)**, pp. 865-882.
[8] S. Camtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 15, no. 2, **(2007)**.
[9] J. Lee and D. R. Stinson, "Deterministic Key Pre-Distribution Schemes for Distributed Sensor Networks", ACM Symposium on Applied Computing 2004, Lecture Notes in Computer Science, vol. 3357, Waterloo, Canada, **(2004)**, pp. 294–307.
[10] R. Blom, "Theory and Application of Cryptographic Techniques", Proceedings of the Eurocrypt 84 Workshop on Advances In Cryptology, **(1984)** Springer, Berlin, pp. 335–8.
[11] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences", Proceedings Of Crypto 92, **(1992)**.
[12] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. Tygar, "SPINS: Security Protocols for Sensor Networks", Proceedings of ACM Mobicom, **(2001)**.
[13] L. Eschenauer and V. D. Gligor, "A Key-Management Scheme for Distributed Sensor Networks", Proceedings of 9th ACM Conference on Computer and Communications Security, **(2002)**.
[14] B. Lai, S. Kim and I. Verbauwhede, "Scalable Session Key Construction Protocol for Wireless Sensor Networks", Proceedings of IEEE Workshop on Large Scale Real Time and Embedded Systems Lartes, **(2002)**.
[15] H. Chan, A. Perrig and D. Song, "Random Key Pre-Distribution Schemes for Sensor Networks", Proceedings of IEEE Symposium on Research in Security and Privacy, **(2003)**.
[16] D. Liu and P. Ning, "Location-Based Pair-Wise Key Establishment for Static Sensor Networks", Proceedings of 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, **(2003)**.
[17] D. Liu and P. Ning, "Establishing Pair-Wise Keys in Distributed Sensor Networks", Proceedings of 10th ACM Conference on Computer and Communications Security CCS'03, **(2003)**.
[18] R. Pietro, L. Mancini, Y. Law, S. Etalle and P. Havinga, "LKHW: A Directed Diffusion- Based Secure Multicast Scheme for Wireless Sensor Networks", Proceedings of First International Workshop on Wireless Security and Privacy, **(2003)**.
[19] W. Du, J. Deng, Y. Han, S. Chen and P. Varshney, "A Pairwise Key Pre-Distribution Scheme for Wireless Sensor Networks", Proceedings of the 10th ACM Conference on Computer and Communications Security, **(2003)**.
[20] S. Zhu, S. Setia and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", Proceedings of the 10th ACM Conference on Computer and Communications Security, **(2003)**.
[21] S. Camtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", Proceedings of the 9th European Symposium on Research Computer Security, **(2004)**.
[22] D. Huang, M. Mehta, D. Medhi and L. Harn, "Location-Aware Key Management Scheme for Wireless Sensor Networks", Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks, **(2004)**, Washington, DC, USA, pp. 29–42.
[23] W. Du, J. Deng, Y. Han, S. Chen and P. Varshney, "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge", Proceedings of the IEEE Infocom'04, **(2004)**.
[24] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment in Sensor Networks", Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, **(2005)**, Miami, Fl, USA, pp. 524–35.
[25] Z. Yu and Y. Guan, "A Robust Group-Based Key Management Scheme for Wireless Sensor Networks", Proceedings of IEEE Wireless Communications and Networking Conference, **(2005)**, New Orleans, LA, USA, pp. 13–7.

[26] J. Jang, T. Kwon and J. Song, "A Time-Based Key Management Protocol for Wireless Sensor Networks", Proceedings of ISPEC, **(2007)**, LNCS 4464, pp. 314–328.
[27] W. Zhang, M. Tran, S. Zhu and G. Cao, "A Random Perturbation-Based Scheme for Pairwise Key Establishment in Sensor Networks", Proceedings of MOBIHOC'07, **(2007)**, Montréal, Québec, Canada.
[28] C. Yu, C. Lu and S. Kuo, "A Simple Non-Interactive Pairwise Key Establishment Scheme in Sensor Networks", Proceedings of the IEEE Secon-09, **(2009)**.
[29] B. Dutertre, S. Cheung and J. Levy, "Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust", Proceedings of Tech. Rep. Sri-Sdl-04-02, **(2004)**, System Design Laboratory.

Authors

Suman Bala received her M.E degree in Computer Science & Engineering from Thapar University, Patiala, India. She had received B-Tech degree from Punjab Technical University, Jalandhar, India. She is pursuing Ph. D from Thapar University, Patiala, India.

Gaurav Sharma received his M.E degree in Computer Science & Engineering from Thapar University, Patiala, India. He had received M. Sc. as well as B. Sc. degree from CCS University, Meerut, India. He is pursuing Ph. D from Thapar University, Patiala, India.

A. K. Verma is currently working as Assistant Professor in the department of Computer Science and Engineering at Thapar University, Patiala in Punjab (INDIA). He received his B.S. and M.S. in 1991 and 2001 respectively, majoring in Computer Science and Engineering. He has worked as Lecturer at M.M.M. Engg. College, Gorakhpur from 1991 to 1996. From 1996 he is associated with the same University. He has been a visiting faculty to many institutions. He has published over 80 papers in referred journals and conferences (India and Abroad). He is member of various program committees for different International/National Conferences and is on the review board of various journals. He is a senior member (ACM), LMCSI (Mumbai), GMAIMA (New Delhi). He is a certified software quality auditor by MoCIT, Govt. of India. His main areas of interests are: Programming Languages, Soft Computing, Bioinformatics and Computer Networks. His research interests include wireless networks, routing algorithms and securing ad hoc networks.